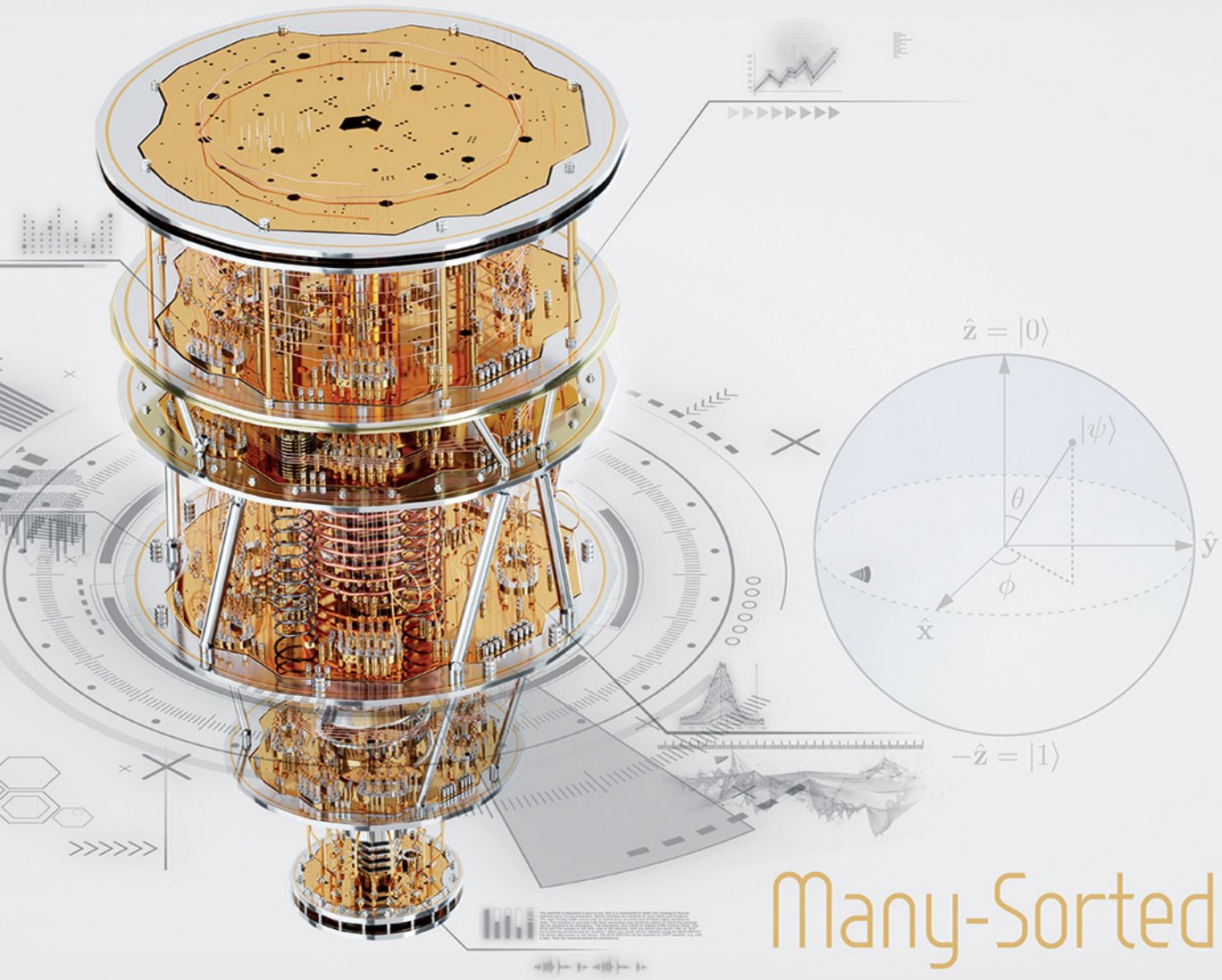


Many-Sorted Algebras for Deep Learning and Quantum Technology

Charles R. Giardina

MORGAN KAUFMANN

MANY-SORTED ALGEBRAS FOR DEEP LEARNING AND QUANTUM TECHNOLOGY

This page intentionally left blank

MANY-SORTED ALGEBRAS FOR DEEP LEARNING AND QUANTUM TECHNOLOGY

CHARLES R. GIARDINA
Lucent Technologies, Whippany, NJ, United States (Retired)

Morgan Kaufmann is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

ISBN: 978-0-443-13697-9

Publisher: Mara Conner
Acquisitions Editor: Chris Katsaropoulos
Editorial Project Manager: John Leonard
Production Project Manager: Selvaraj Raviraj
Cover Designer: Matthew Limbert

Typeset by MPS Limited, Chennai, India

Dedication

To my everything, Betty

This page intentionally left blank

Contents

List of figures

Preface

Many-sorted algebra (MSA) provides a rigorous platform for describing and unifying various algebraic structures in several branches of science, engineering, and mathematics. However, the most natural application for the use of this platform is in all areas of quantum technologies. These include quantum physics, quantum mechanics, quantum information theory among the most recent, quantum computing, quantum neural nets, and quantum deep learning. Indeed, in all quantum disciplines, there exists an abundance of algebraic underpinnings and techniques. Several of these techniques are directly applicable to machine learning and are presented herein. In particular, with the current interest in quantum convolutional neural networks, understanding of basic quantum becomes all the more important. Although analytical, topological, probabilistic, as well as geometrical concepts are employed in many of these disciplines, algebra exhibits the principal thread. This thread is exposed using the MSA.

A fundamental setting of Hilbert space over a complex field is essential in all of quantum, while machine learning deals predominantly with the real field. Both a global level and a local level of precise specification are described in the MSA. Indeed, characterizations at a local level due to distinct carrier sets may appear very different, but at a global level they may be identical. Banach* algebras as well as Hilbert spaces are basic to these systems. From a local view, these algebras may differ greatly. For instance, the Banach algebra-type bilinear multiplication operation in the neural network might involve an affine map or it could be convolution. However, in quantum systems, the bilinear operation often takes additional forms. These include point-wise multiplication, function composition, Lie or Poisson brackets, or even a concatenation of equivalence classes of paths in homotopy.

Theoretical as well as practical results are provided throughout this text. Hilbert space rays acting as states, as well as an in-depth description of qubits, are explained and illustrated. Qubits form a center stage in quantum computing and quantum machine learning. Unitary operators forming a group are employed in state transitions and are described in the MSA as Hilbert and Lie groups. Parameters within unitary operators allow optimization in quantum computing applications. Concurrently, C* algebras described in the MSA embrace the structure of observables. In all cases, the MSAs are most useful in illustrating the interplay between these and the various other algebraic structures in quantum and machine learning disciplines.

Charles R. Giardina

This page intentionally left blank

Acknowledgments

I am thankful to Ed Cooke for reading the preliminary chapters and to the reviewers for their constructive improvements. Thanks are also due to the Elsevier editorial staff for their assistance and helpful guidance throughout. Finally, I am most appreciative of knowing and learning from the three great mathematicians/engineers and all one-time Bell Telephone Laboratories' colleagues: David Jagerman, Charles Suffel, and Frank Boesch.

This page intentionally left blank

CHAPTER

1

Introduction to quantum many-sorted algebras

1.1 Introduction to quantum many-sorted algebras

This chapter begins by mentioning several algebraic structures described in the later sections of the text that will be embedded into a version of the many-sorted algebra. This is followed by a description, as well as an illustration of the many-sorted algebra methodology. A global view involving the MSA is given using polyadic graphs consisting of nodes with many tailed directed arrows. The general field structure is described in Section 1.1.4, in terms of the MSA. This is followed by further algebraic structures in quantum and machine learning. Specific quantum and machine learning fields are presented along with general Hilbert space conditions that underly all quantum methodology. Time-limited signals are developed under inner product space conditions. These signals are basic constructs for convolutional neural networks. Kernel methods, useful in both quantum and machine learning disciplines, are presented. In later chapters, kernel methods are shown to be a fundamental ingredient in support vector machines. This chapter ends with a description and application of R modules. These structures have an MSA description almost identical to a vector space structure.

1.1.1 Algebraic structures

Throughout quantum, an extremely wide variety of algebraic structures are employed, beginning with the most fundamental canonical commutation relations (CCR) to methods for solving elliptic curve cryptography and building quantum convolutional neural networks. It is the purpose of this text to provide a unification of the underlying principles embedded within these algebraic structures. The mechanism for this unification is the many-sorted algebra (MSA) (Goguen and Thetcler, 1973). The MSA can be thought to be an extension of universal algebra, as in Gratzer (1969). Here, varieties of algebraic structures are described in a most generalized sense with morphisms showing correspondence between objects. The underlying characterization of the many-sorted or many types of algebraic concepts in quantum disciplines is captured simultaneously through rigorous

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00019-9

specification as well as polyadic graphs within the MSA. The present work is inspired by Birkhoff and Lipson (1970) and their heterogeneous algebras, as well as (Goguen and Meseguer, 1986) remarks on the MSA.

Both a global level and local level of precise specification are presented using the MSA. The MSA is essential for a better understanding of quantum and its relationship with machine learning and quantum neural network techniques. The very concept of Hilbert space from the beginning axioms is detailed in a precise but high-level manner (Halmos, 1958), whereas underlying fields for quantum and machine learning are very specific. In quantum, this field is almost always complex; sometimes real numbers or even quaternion numbers are utilized. However, in machine learning, it is the field of mainly the reals that is employed. This is particularly true with support vector machine applications.

In general, the quantum Hilbert space could be finite dimensional, it could consist of kets and bras, and it could be a tensor product of similar Hilbert spaces or infinite dimensional as is L^2 or l^2 (Halmos, 1957). All of these structures will precisely be explored at a local level. This is again evident in specifying the Gelfand-Naimark-Segal (GNS) construction relating a C* algebra to a Hilbert space (Gelfand and Naimark, 1943; Segal, 1947). From a practical viewpoint, Hilbert spaces of qubits are described for use in a quantum computer (Feynman, 1986). Other applications include qubits in quantum neural networks and quantum machine learning.

1.1.2 Many-sorted algebra methodology

To begin describing the MSA methodology, the set consisting of the sorts of objects must be specified. For instance, the term scalar may be an element of this set. Although the term scalar is generic, it might refer to elements from a field such as the real or complex numbers. However, it might also represent a quaternion that is an element from a skew or noncommutative field. Importantly, for each sort, there are carrier sets. It is these sets that uniquely identify the precise type of elements in question. For instance, very different carrier sets are used for the real field, the complex field, the rational field, or a finite field that is employed in cryptography.

Once the sorts are declared, operational symbols must also be given. They are organized as elements within specific signature sets. These sets are used in identifying common attributes among symbols such as their arity. Operational symbols denote the inter- and intrasort mappings like symmetrization, annihilation, creation, as well as elementary operations: addition, multiplication, inversion, and so on.

The actual operators utilized in these mappings involve specified carrier sets that correspond to the sorts. This is performed at a lower view. Each operator employs elements from designated carrier sets as operands in the domain. This is true for their codomain as well. The operator names within signature sets are enumerated along with the algebraic laws, rules, equational identities, or relations which they must obey. The laws or equational identities include commutation rules, associative laws, distributive laws, nilpotent rules, and various other side conditions or relations necessary for rigorous specification.

A useful global view involving the MSA is given using polyadic graphs consisting of nodes with many tailed directed arrows (Goguen and Meseguer, 1986). All entities of the arrow are labeled. Each node is denoted by a circle inscribed with a specific sort. The arrows have operator names attached and are as declared in their signature set. The number of tails in the arrow corresponds to the arity of the operator in question. Operators of arity zero have no tails and are labeled using the name of special elements of the sort. These include zero, one, and identity element, as well as top or bottom. The tails of an arrow are emanating from the specified domain sorts coming from the appropriate signature sets. The single head of the arrow points to the sort of codomain for the operator. In short, the polyadic diagram provides a visual description of the closure operations needed in describing the algebraic structure.

Partial operators are included in the MSA. This is similar to what is done in partial universal algebra. However, special notations are employed for operations not defined on the entire denoted domain sort. Much of this notation will be given later. The inclusion of domain-dependent operators is essential in quantum since even the position and momentum operators are unbounded. Infinite-dimensional Hilbert space necessities explicit domain declaration as well as closure conditions. A more basic instance of an operator not fully defined will be given right now. Here, the multiplicative inverse in a field is defined for all values except for zero, and thus it is a partial operator. However, this operator exists in the MSA. Moreover, a dashed arrow with a single tail is utilized in the polyadic graph description in this case.

1.1.3 Global field structure

Corresponding to the field structure, only the single sort SCALAR is needed. Several signature sets exist. They are organized by the arity. Arity refers to the number of operands or arguments for operators within the given signature set. Arity also refers to the number of tails of a polyadic arrow. For binary operators, unless specified otherwise, they should utilize both arguments in either order. There is no restriction to which argument comes first.

Binary operation: {ADD, MULT} each maps SCALAR × SCALAR → SCALAR
Unary operation: {MINUS, INV} each maps SCALAR → SCALAR
Zero − ary operation: {ZERO, ONE} these are special elements of the sort SCALAR.

Note that even though INV is a partial function name, it is contained in the same signature set as MINUS; both are unary operator names. Fig. 1.1 provides an illustration of a high-level interpretation of an algebraic field. This graph indicates the closure operations. For instance, the ADD implies that two values from SCALAR are combined to give another value of SCALAR, whereas MINUS takes a single value of SCALAR and yields another such value. The arrow pointing from ZERO to SCALAR indicates that there has to be an element in the field whose name is ZERO. The same is true for ONE. As in universal algebra, the number of operational names of a specific arity is often listed by a finite

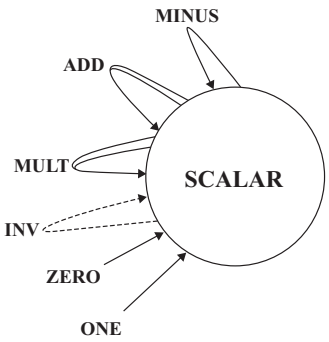


FIGURE 1.1 Polyadic graph for the field structure.

sequence of nonnegative integers. For a general field structure, the arity sequence is given as follows: (2, 2, 2). Indeed, the first entry specifies the number of zero-ary operations; here it is 2, while the next entry is for the number of unary operations = 2 and the final entry is the number of binary operations = 2. The listing procedure is similar to the method of recording the number of fermion or boson occupational numbers in Fock space. This space will be described in later sections.

The equational identities or laws for a field are given below. Here, for convenience, we denote the sort by representative symbols and all the operational names by suggestive symbols.

SCALAR by a, b or c.
ADD by +
MULT by ·
MINUS by -
INV by /
ZERO by 0
ONE by 1

The equational identities, laws, or constraining equations for a field are as follows:

1) Associative for addition: $(a + (b + c)) = ((a + b) + c)$
2) Zero law: $0 + a = a + 0 = a$
3) Minus law: for any, a there is $-a$, where $a - a = -a + a = 0$
4) Commutative law for addition of all elements: $a + b = b + a$
5) Associative law for multiplication: $(a \cdot (b \cdot c)) = ((a \cdot b) \cdot c)$
6) Distributive laws: $a \cdot (b + c) = a \cdot b + a \cdot c$; $(a + b) \cdot c = a \cdot c + b \cdot c$
7) One law: $1 \cdot a = a \cdot 1 = a$
8) Partial inverse law, exclude 0: for any a, there is $1/a$ where: $a \cdot 1/a = (1/a) \cdot a = 1$
9) Commutative law for multiplication: $a \cdot b = b \cdot a$.

An example of an abstract field F_3 will be given to illustrate the closure operations, which is the essence of Fig. 1.1. Also illustrated are the nine, equational constraints listed earlier. The example is important in the preparation for the development of elliptic curve cryptography and Shor's quantum algorithm described in a later chapter.

Example 1.1:

Consider the carrier set for SCALAR to be the set X = {0, 1, 2}. Operations corresponding to those named in the signature sets are defined as modular three. The following tables provide the binary ADD, MULT, and the unary operation MINUS, as well as the partial unary operation INV; these are listed in order as follows:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

x	−x
0	0
1	2
2	1

x	1/x
1	1
2	2.

To use the first two tables to find the elements to the left and above for which the binary operation is to be performed, the result is located in the row and column to the right and below, respectively. For the two tables to the right, use the first column; then the unary operation can be read to the right of the desired element.

The equational identities all hold. To show (1) all possible values of a, b, and c must be utilized. Here, there are 27 combinations, but only a single instance is illustrated next.

1) Associative for addition: (2 + (1 + 2)) = (2 + 0) = 2; also ((2 + 1) + 2) = 0 + 2 = 2
2) Zero law: From the + table, 0 on the left or above when added to x gives x
3) Minus law: From the—table, for example, 1 + 2 = 2 + 1 = 0, −2 = 1
4) Commutative law for addition: The + table is symmetric about the main diagonal
5) Associative law for multiplication: (2 · (1 · 2)) = (2 · 2) = 1; also ((2 · 1) · 2) = 1
With the first four identities holding, this shows that the additive structure is an abelian group. Additionally, it is an instance of a cyclic group with three elements. The addition wraps around 2 + 1 = 0. As in the associative laws, the distributive laws actually need all 27 arrangements for full validation. However, as before, only one case is illustrated next.
6) Distributive laws: (2 · (1 + 2)) = (2 · 0) = 0; also (2 · 1) + (2 · 2) = 2 + 1 = 0
7) One law: From the · table, the 1 on top or to the left multiplying x gives x
8) Partial inverse law, exclude 0, from the last table 1/1 = 1, and 1/2 = 2
9) Commutative law for multiplication: The table · is symmetric about the main diagonal.

Since all the equational identities hold along with the closure operations, this shows that the structure F_3 is a field. The field is called a finite field or a Galois field.#

1.1.4 Global algebraic structures in quantum and in machine learning

To conserve space and take advantage of the general field structure earlier, we mention important substructures of a field. The listing attempts to go from the most general structure, a groupoid, to the most restrictive, a field. All structures utilize as their sort SCALAR and involve operational names from signature sets provided for the field. Moreover, most of the following algebraic structures require some of the equational identities, (1) through (9). These are listed earlier, providing the global description of an algebraic field. Finally,

the polyadic graph for these structures is the same as that for a field, but possibly with some arrows removed. Many of the forthcoming structures often appear in quantum disciplines and will be applied in subsequent sections. The specifics below should act as a reference to the global definition of these structures.

A groupoid is a structure with only a single signature set consisting of ADD with no constraints. A groupoid satisfying constraint (1) is a semigroup. When there is also a ZERO along with constraint (2), the semigroup is called a monoid. If in addition there is a MINUS and (3) holds, then a monoid is a group. The group is called Abelian when (4) holds. When MULT also exists and (5) and (6) hold, the Abelian group is called a ring. If ONE also exists along with (7), the ring is called a ring with identity. A ring in which (9) holds is said to be a commutative ring. When ONE exists and (7) and (9) hold, the ring is a commutative ring with identity or with unity. A commutative ring with unity is said to be an integral domain when there does not exist divisors of zero. Divisors of zero occur when the product of two nonzero elements equals ZERO. A skew-field arises when a ring with identity also has an INV obeying (8); this structure is also called a division ring. When (9) also holds, the skew-field is said to be a field.

Illustrations of many of these structure are described in the subsequent chapters, for instance, Lie groups and Lie algebras; also the quaternions provide an instance of a division ring. Below is an important example of a unital commutative ring that is not a field. It is a structure that is easy to understand, but this carrier set is of critical importance for use in R-modules. It will be seen in a subsequent section that fields are to vector spaces, as rings are to R-modules.

Example 1.2:

Consider the carrier set of all the integers Z. If the usual addition, negation, and ZERO are employed, then this structure becomes an abelian group. If the usual multiplication and ONE are introduced, along with all the equational identities specified above for a field, except (#8), then this structure is a unital commutative ring. Additionally, the polyadic graph in Fig. 1.1, modified for a ring structure, might have the dotted partial operation INV arrow removed. However, it might not, since in the integers the numbers one and minus one do have inverses.#

All group and group-like structures mentioned earlier are additive group or group like. In quantum and in machine learning, many of these corresponding structures are similar algebras. For instance, they are often multiplicative group or multiplicative group like.

Any and every field can be described in the manner specified earlier. This was the high-level or big picture. Again, the sort SCALAR and these signature sets hold true for the rational field, the real field, the complex field, or any Galois or finite field, as illustrated in the last section. Now, two additional specific fields will be identified.

1.1.5 Specific machine learning field structure

To obtain the real field (R) underlying machine learning, the carrier set relating to the sort SCALAR are the real numbers. It provides the actual lower, in-depth view. In addition, for each operator name within a signature set, an actual operator or function of the

same arity is defined. All the equational constraints and laws hold true using these elements. In particular, ZERO in this case is 0, and for any real number r, $r + 0 = 0 + r = r$. Also ONE is the number 1, and $1 \cdot r = r \cdot 1 = r$. Finally, the inverse s, of any real number r, other than 0 can be found, $s = 1/r$.

1.1.6 Specific quantum field structure

To obtain the complex field (C) underlying the Hilbert space in most quantum situations, the sort SCALAR refers to the complex numbers. In addition to each operator name within a signature set, an actual operator or function of the same arity must be defined. The carrier set here is the complex number system. It provides the actual lower, in-depth view. The actual carrier set for SCALAR is {$x + i \cdot y$ also written as $x + iy$ or $x + yi$, such that x and y are now real numbers and i is a nonreal number; it is a symbol having the property that $i^2 = -1$}. Moreover, the plus sign is just a character holding the two entities together. The closure operations provided in Fig. 1.1 must be rigorously specified. For instance, for two complex numbers, $v = a + i \cdot b$ and $w = c + i \cdot d$, ADD (v, w) = $(a + c) + i \cdot (b + d)$. There are two different plus signs in the addition formula. To make things worse, we will write ADD (v, w) = $v + w = (a + c) + i \cdot (b + d)$. Now, there are three uses of the plus sign. However, not to go crazy with notation, we will continue with this practice. Sometimes different notations such as $+_1$, $+_2$, and $+_3$ are used to make things clearer. Indeed, in later chapters, Hilbert spaces of linear mappings employ all three plus signs. One last abuse of notation is for the zero-ary element ZERO use $0 + i \cdot 0 = 0$. A quicker explanation of the complex field now follows.

All the equational constraints and laws hold true using these elements. In particular, only the following two laws are mentioned for $z = (x + i\ y)$:

3) Minus law: MINUS $(x + i\ y) = (-x - i\ y)$.
8) Partial inverse for non (0 + 0 i): INV $(x + i\ y) = x/(x^2 + y^2) - i\ y/(x^2 + y^2)$.

Letting $z = x + iy$, then the real part of z is denoted by Re(z), and it is x. Likewise, the imaginary part of z is Im(z) and it is y. Note that they are both real valued. A very important operation in the complex field is conjugation. It is an operation that cannot be derived in terms of the other operations that are referred to in the signature sets. Conjugation has operator symbol CON. When applied to a complex number, it negates the imaginary part. The actual operation is *, and so the abusing notation is as follows: CON(z) = CON $(x + i\ y) = (x - i\ y)$. More precisely, $z^* = (x + i\ y)^* = (x - i\ y)$. Moreover, the operation of conjugation is an involution; therefore two applications of conjugation result in the original value. Two applications act like the identity operation. Thus, it follows that $(z^*)^* = ((x + i\ y)^*)^* = (x + i\ y) = z$. The absolute value of a complex number z is the square root of the number multiplied by its conjugate. Equivalently, $|z|^2 = z^*z = z\ z^*$. Also note that the real part of z is Re(z) = $(z + z^*)/2$ and the imaginary part of z is Im(z) = $(z - z^*)/2i$. Both of these quantities are real valued. All the aforementioned properties are needed in subsequent examples involving inner products as well as in describing adjoint operations. Finally, the polar form for any complex value $z = x + iy$ can be written as $z = r\ e^{i\theta}$, where $r = (x^2 + y^2)^{1/2}$ and $\theta = \arctan(y/x)$. Mentioned previously, the square root should always be interpreted as yielding a nonnegative result.

1.1.7 Vector space as many-sorted algebra

A vector space consists of two distinct sorts of objects. These are SCALAR, as in the field structure, and the second sort VECTOR. Refer to Fig. 1.2; in this diagram, both sorts are illustrated. However, only those operation names that exclusively involve the sort SCALAR are not displayed. That is, the many-sorted polyadic graph arrows from Fig. 1.1 are not repeated. Corresponding to a vector space structure, the signature sets are organized not only by the arity of operations but also by their types. This is because several operations of the same arity have mixed types of inputs or outputs. For instance, it can be seen later in the figure that the operational names of arity two, that is, V-ADD and S-MULT, will have two distinct signature sets. Specifically, V-ADD takes two VECTORS and returns a VECTOR, whereas S-MULT takes a SCALAR and a VECTOR and returns a VECTOR.

This results in an arity sequence: (1, 1, 2 (1, 1)) for a vector space. For the MSA, the arity listing is as in universal algebra. It identifies in order the number of (Zero-ary, Unary, Binary, Trinary, ... N-ary) operational names. However, different signature sets of the same arity have special rules in MSA. The total quantity of that arity in the arity sequence is followed by the number of each distinct signature set of that arity. So the two ones after the 2 in the arity sequence show that there are two distinct operations of arity two, all with different domains or codomains. The actual signature sets for a vector space starting with higher arity and decreasing in order are the following:

$$\begin{aligned}
&\text{Binary operation } \{V-ADD\},\ V-ADD \text{ maps } VECTOR \times VECTOR \to VECTOR\\
&\qquad\qquad\qquad\quad \{S-MULT\},\ S-MULT \text{ maps } SCALAR \times VECTOR \to VECTOR\\
&\text{Unary operation } \{V-MINUS\},\ V-MINUS \text{ maps } VECTOR \to VECTOR\\
&\text{Zero}-\text{ary operation } \{V-ZERO\},\ V-ZERO - \text{is a special element of the sort } VECTOR.
\end{aligned}$$

As previously mentioned, the binary operators utilize their operands in either order. For instance, for scalar a and vector v, S-MULT (a; v) = a • v = v • a.

Note that only three signature sets mentioned earlier have operational names that utilize sort VECTOR exclusively. That is, these three names associate operators with the domain and codomain of sort VECTOR. These are of arity 0, 1, and 2. The corresponding operators within these sets alone describe the additive abelian vector group

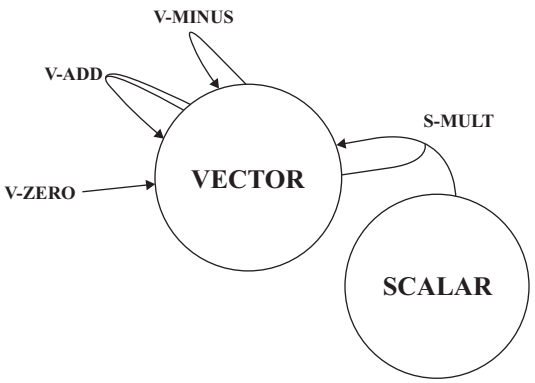


FIGURE 1.2 Vector space described as many-sorted algebra.

within the vector space. Here, equational constraints (1) − (4) mentioned must also hold. The arity sequence for this additive group is (1, 1, 1).

The equational identities or laws for a vector space are given below. This is followed, in the next section, by additional equational identities needed for an inner product or Hilbert space. Again for convenience, we denote the sorts by representative symbols and all the operational names by suggestive symbols.

SCALAR by a, b or c.
VECTOR by u, v or w
ONE by 1
V-ADD by +
V-MINUS by −
V-ZERO by 0
S-MULT by ·

1) Associative for vector addition: $(u + (v + w)) = ((u + v) + w)$
2) Zero vector law: $0 + v = v + 0 = v$
3) Minus vector law: $v + (-v) = -v + v = 0$
4) Commutative vector law for addition: $u + v = v + u$
5) One law: $1 \cdot v = v \cdot 1 = v$
6) Distributive law: $a \cdot (u + v) = a \cdot u + a \cdot v$
7) Distributive law: $(a + b) \cdot u = a \cdot u + b \cdot u$
8) Associative law: $(a \cdot b) \cdot v = a \cdot (b \cdot v)$

These eight laws describe any vector space in generality.
An interesting example of a structure that fails to be a vector space is given next.

Example 1.3:

Let the carrier set for SCALAR be all the real numbers R, with the usual real field structure. However, let the carrier set for VECTOR be the positive real numbers $V = R+ = \{x, \text{ such that } 0 < x < \infty\}$, with usual multiplication and division. In this application, in place of vector addition, multiplication of the vectors is used. That is, multiplication of positive real numbers is employed. Since the product of two positive real numbers is a positive real number, this binary operation is closed. The multiplication operation in this case is valid. For the unary minus operation, the inversion operation is substituted. Again, this operation is also closed, since for any positive real number denoting a vector the reciprocal is also a positive real number. In place of the zero vector, the number one is used in this structure. So V-ZERO is the number one.

Finally, the scalar multiplication involving vectors must be described. The important criterion again is that this operation is closed; that is, it must satisfy the closure operations inherent in the polyadic graph in Fig. 1.2 for vector space. The actual operation in this case is performed in two steps. First, form the product of the scalar real value with the positive real value vector. Then next, use this product as an exponent of the power of e. Upon applying this two-step operation, again the result is always a positive real number. Consequently, the operation is closed, and a vector is again obtained. The operations described earlier are given again, but in a more formal manner.

First:

Denote SCALARS by a, b, and c; these are all real numbers.
Denote VECTORS by u, v, and w; these are all positive real numbers.

Replace the operation name by the actual carrier set operation:

V-ADD (v,w) by $v \cdot w$
V-MINUS (v) by $1/v$
V-ZERO by 1
S-MULT (a; v) = $e^{(av)}$, so a and v are multiplied and become an exponent of e.

Identifying the operational names whose signature sets only include sort VECTOR results in an Abelian group structure. Notice that all the equational identities hold using the specified carrier set, where the name ADD refers to multiplication:

1) Associative for vector addition: $(u \cdot (v \cdot w)) = ((u \cdot v) \cdot w)$.
2) Zero vector law: $1 \cdot v = v \cdot 1 = v$.
3) Minus vector law: $v \cdot 1/v = (1/v) \cdot v = 1$.
4) Commutative vector law for addition: $u \cdot v = v \cdot u$.

Thus, an Abelian group structure is verified. However, the structure does not satisfy all the equational identities that define a vector space. In fact, it does not satisfy all the following side conditions. So, e raised to a real power is always a positive real number and is itself a vector in this space, and closure exists. However, all (5) − (8) equational identities must also hold for a vector space structure.

5) One law: $e^{(1v)} = e^{(v1)} = e^{(v)}$, this holds.
6) Distributive law: $e^{(a(uv))}$, not equal to $e^{(au)}e^{(av)} = e^{a(u+v)}$ and doesn't hold in general.
7) Distributive law: $e^{((a+b)u)} = e^{(au)}e^{(bu)} = e^{(au+bu)}$, this holds.
8) Associative law: $e^{(ab(v))}$, not equal to exp(a $e^{(bv)}$) and doesn't hold in general. #

The next example utilizes carrier sets exactly the same as in the previous example, but only a change is made in the definition of scalar multiplication.

Example 1.4:

For the same conditions as in the last example, but this time, the only change is to let the scalar multiplication be redefined. So the carrier set for SCALAR is again R. The carrier set for VECTOR is again R^+, all the positive real numbers.

Denote SCALAR by a, b, and c, all real numbers.
Denote VECTOR by u, v, and w, all positive real numbers.
Replace the operation name by the actual carrier set operator:

V-ADD (v,w) by $v \cdot w$
V-MINUS (v) by $1/v$
V-ZERO by 1
S-Mult (a; v) = v^a.

The last operation is the change from the previous example. In the present case, the vector is raised to the scalar power. Since a positive real number when raised to any real

power is itself positive, this verifies the closure condition. Thus, the vector space diagram, that is, Fig. 1.2, is valid, but still all the equational identities must also hold for this structure to be classified as a vector space.

So to verify that this is a vector space, note that all the following do hold:

1) Associative for vector addition: $(u \cdot (v \cdot w)) = ((u \cdot v) \cdot w)$.
2) Zero vector law: $1 \cdot v = v \cdot 1 = v$.
3) Minus vector law: $v \cdot 1/v = (1/v) \cdot v = 1$.
4) Commutative vector law for addition: $u \cdot v = v \cdot u$.
5) One law: $v^1 = v$.
6) Distributive law: $(vw)^a = v^a\ w^a$.
7) Distributive law: $v^{(a+b)} = v^a\ v^b$.
8) Associative law: $v^{(ab)} = (v^a)^b$.#

In terms of vector spaces, two distinct carrier sets have been defined so far for sort SCALAR: They are the real (R) and the complex (C) number fields. For these cases, a vector space is said to be real whenever the scalar field is R. It is said to be complex whenever the scalar field is C. Accordingly, the operation whose name is S-MULT must take a vector and multiply it by a scalar and obtain a vector in the designated carrier set of sort VECTOR. In a sense, the carrier set of sort SCALAR governs the nature of the vector space.

Example 1.5:

A most simple real vector space is when the carrier sets for VECTOR and SCALAR are both equal to the reals R. Here, vectors can be thought of as arrows on the x axis with their tails at the origin. While scalar multiplication is used to stretch or contract these arrows, a negative scalar will reverse the arrow by one hundred eighty degrees and scalar zero would yield the origin.#

Example 1.6:

Another real vector space is when the carrier sets for VECTOR are the complex numbers C, and the SCALAR are the reals R. Here vectors can be thought of as arrows on the $x - y$ plane with their tails at the origin. Again, scalar multiplication will only elongate or shorten them. The arrows will become the origin when the scalar zero is employed. While using negative numbers, for instance, using -1, a rotation of 180 degrees is applied to a vector.#

Example 1.7:

A complex vector space occurs when both carrier sets for VECTOR and SCALAR are both equal to the complex numbers. As in the previous example, vectors can be thought to be in the $x - y$ plane with tails at the origin. When scalar multiply uses a complex number: $z = x + i\ y = r\ e^{i\theta}$, the nonzero vector will elongate or shrink by $r = |z|$ and rotate by an angle of θ.#

1.1.8 Fundamental illustration of MSA in quantum

A high view involving the MSA is described later for the fundamental setting of a separable Hilbert space over the complex field or real field. The algebraic underpinnings are those of a vector space, along with the inner product operation and corresponding equational constraints. These define an inner product space. Specifically, equational identities (1) − (8) from the previous section must hold in addition to (9) − (11) specified later.

The more topological notions such as those needed for describing tangent bundles, other bundles, as well as Lie groups and Lie algebras will be specified in the MSA. However, for completeness, we quickly describe the high-level topological or analytical foundations for an inner product space to become a Hilbert space indicated earlier. To begin, separable means there exists a countable dense subset within the Hilbert space. This allows for the introduction of a Schauder basis, thus creating infinite dimensional Hilbert spaces with operations similar to those with a Hamel basis. The latter basis is utilized in all of finite dimension vector spaces. Expansions of vectors in terms of Schauder basis elements become almost identical to the finite dimensional situation. Finally, basic to a Hilbert space is that every Cauchy sequence converges in norm; this is the extra criteria for an inner product space to become a Hilbert space. See also Appendix A.1 for an in-depth description of convergence and completeness. Throughout the document, a separable Hilbert space is assumed, except when specifically stated otherwise. In finite dimensional real and complex vector space situations, these topological and analytical properties always hold.

The set of sorts for describing an inner product space or a Hilbert space is {SCALAR, VECTOR}. As in a vector space, each element of the set of sorts is depicted as a circular node within the polyadic graph. Each element within a signature set is denoted by an arrow in the polyadic graph. The many-tailed arrow is labeled with the name of the specific element of the signature set. The MSA description begins with the underlying scalar field global structure. As mentioned before, it is the general setting for both the real numbers, basic to machine learning, and the complex numbers, fundamental in Hilbert space quantum theory. Additionally, it is the underlying structure used in finite field cryptography.

The actual signature sets for an inner product space or Hilbert space, starting with higher arity and decreasing in order, are given below. They are the same as for a vector space, but it includes an additional operator name, IN-PROD:

Binary operation {V − ADD}, V − ADD maps VECTOR x VECTOR → VECTOR
{S − MULT}, S − MULT maps SCALAR x VECTOR → VECTOR
{IN − PROD}, IN − PROD maps VECTOR x VECTOR → SCALAR
Unary operation {V − MINUS}, V − MINUS maps VECTOR → VECTOR
Zero − ary operation {V − ZERO}, V − ZERO − is a special element of the sort VECTOR.

These operator names are illustrated in Fig. 1.3.

The arity sequence for an inner product space or a Hilbert space is therefore (1, 1, 3(1, 1, 1)). Note that all three binary operators have either different inputs or different outputs. As mentioned previously when the completeness axiom holds (every Cauchy sequence

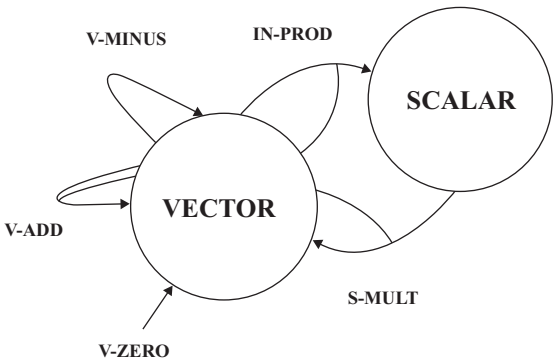


FIGURE 1.3 Inner product or Hilbert space.

converges in norm), the inner product space also becomes a Hilbert space. The next three equational constraints when true make the vector space an inner product space, but first,

Use a and b as SCALAR
Use u, v, and w as VECTOR
Denote the IN-PROD by $< | >$ or by $<, >$

9) Positive definite: $< v | v >$ must be greater or equal to 0 and $= 0$ iff $v = 0$.
10) Conjugate symmetric: $< v | w > = < w | v >^*$, where * is the conjugate operation
11) Conjugate bilinear: $\langle a \cdot (u + v)|w\rangle = a^* \cdot \langle u|w\rangle + a^* \cdot \langle v|w\rangle$

$$\langle u|b \cdot (v + w)\rangle = b \cdot \langle u|v\rangle + b \cdot \langle u|w\rangle.$$

The vector norm of v, denoted $\|v\|_2$ and induced by the inner product, is given by the square root of $< v | v >$. Equivalently, $\|v\|_2^2 = \langle v|v\rangle$. Convergence of sequences and Cauchy criteria in a Hilbert space are described with reference to this norm. It should be mentioned that the conjugate bilinear law given in the inner product identity (11) above is the one usually used in physics and always used in this document. That is, the first argument in the inner product is conjugate linear. In mathematics, usually the second equality in (11) employs scalar conjugation, not the first equality. For machine learning, conjugation is of lesser importance because the scalars are most often real numbers. In this case, the conjugate of a scalar is itself.

1.1.9 Time-limited signals as an inner product space

Consider all real-valued digital time-limited signals or functions f in a subset of R^Z. This is the carrier set corresponding to VECTOR, and R^Z means that these functions have domain equal to all the integers and result in a real value, f: $Z \to R$. Being time-limited means that f can have nonzero values only on some finite subset of the integers Z. These functions have finite support, and the space of all these functions will be denoted by A. Note that V-ZERO corresponds to the 0 vector in A. Convolutional neural nets (CNNs) employ time-limited signals, both as raw data streams and as filtering signals. Applications include one-dimensional sound, text, data, time series, as well as EKG and ECG signal classification (Ribeiro et al., 2020).

Bound vectors are convenient representations for f in A (Giardina, 1991). Any nonzero signal in A will be nonzero at a smallest integer n. In this case, the representation for the function f is $(f(n)\ f(n+1)\ \ldots\ f(n+k))_n$, where f(n + k) is the final value for which f is nonzero. A bound vector can contain all zeros, but from a practical point of view, the ZERO element, 0, will always be employed. By definition, the convention here is that k is a nonnegative integer. The subscript n is a pointer to the location in N of the first nonzero element in f. Usual point-wise addition for bound vectors corresponds to the name V-ADD. The plus sign +, as well as V-ADD itself, will also be employed for the actual point-wise addition. Previously mentioned, V-ZERO denotes the bound vector 0 in A. For f mentioned earlier, it is such that 0 + f = f + 0 = f.

So, for g also in A and not identically equal to zero, then $g = (g(m)\ g(m+1)\ \ldots g(m+j))_m$ and $\text{V-ADD}(f,g) = (f(p) + g(p)\quad f(p+1) + g(p+1)\ \ldots\ f(p+q) + g(p+q))_p$. So, when both f and g are not identically equal to zero, p = min(m,n). In this case, p + q is the largest integer value for which f or g is nonzero. Also, p + q = max(n + k,m + j). S-MULT in this context is just point-wise multiplication by a real number r. Thus, $\text{S-MULT}(r; f) = f \cdot r = r \cdot f = (r \cdot f(n)\quad r \cdot f(n+1)\ \ldots\ r \cdot f(n+k))_n$. Referring to Section 1.7, all the equational identities for a vector space hold true for A. Again, the sum or scalar product of a bound vector might be all zeros. In this case, use ZERO element.

For any f in A, the cozero set is defined to be the set of all points in Z for which f is nonzero. It is the complement of the set where all zeros appear and it is abbreviated as COZ(f). For the function 0, COZ(0) is the empty set. For all other signals f in A, COZ(f) has finite cardinality. Time-limited digital signals f and g in A form an inner product space with a usual ordered dot-type product on the intersection of COZ(f) and COZ(g). For the empty set, zero should be used. The conjugate of a real number is itself; accordingly, only positive definite criteria must be shown. Here, $< f, f >$ is the sum of squared terms. Therefore, the inner product is greater or equal to zero and will equal zero only for f = 0. Thus, the inner product space criteria from the previous section hold for the class of bound vectors.

Example 1.8:

Consider the two signals f and g in A given by bound vectors: $f = (3\ 2\ 1)_0$, and $g = (2{-}1)_0$. Then COZ(f) = {0, 1, 2} and COZ(g) = {0, 1}. Here, the dot or inner product is $< f, g > = 6{-}2 = 4$. Also, $< f, f > = 14$, and $< g, g > = 5$. Finally, V-ADD(f, g) = f + g = $(5\ 1\ 1)_0$, V-MINUS(f) = −f = $(-3\ -2\ -1)_0$, and S-MULT(f; 1.5) = 1.5 $(3\ 2\ 1)_0 = (4.5\ 3\ 1.5)_0$.#

From the aforementioned example, it is seen that the algebraic operations involving bound vectors are somewhat identical to the usual vector space operations on real-valued vectors. The primary difference is in the subscript at the end of the vector specifying the location of the first nonzero element. In any case, the vector norm of f, denoted $\|f\|_2$, induced by the inner product, is given by the square root of $< f|f >$. The square root will always use a radical 1/2 and will denote a nonnegative quantity, unless specified differently. Equivalently, $\|f\|_2^2 = \langle f|f\rangle$. Convergence of sequences and Cauchy criteria in a Hilbert space are described with reference to this norm. To see that this inner product space A is

not a Hilbert space, use a sequence of bound vectors: $f_n = (1\ 1/2\ 1/3 \ldots 1/n)_0$, $n = 1, 2, \ldots$. Notice that for $n > m$ both are larger than some positive integer N, then, for instance, using the two norms in l^2: $\|f_n - f_m\|_2^2 = 1/(n)^2 + \ldots + 1/(m+1)^2$. Now, for N large enough, this quantity will be arbitrarily small. This shows the sequence of bound vectors $f_{k,}$ forms a CS. However, in the limit $f_k \to (1\ 1/2\ 1/3 \ldots)_0$, which is not a bound vector. Thus, A is not complete, and it is only an inner product space.

1.1.10 Kernel methods in real Hilbert spaces

Kernel methods are closely related to reproducing kernel Hilbert spaces (RKHSs), which is a topic in a later chapter. RKHSs are special Hilbert spaces where the elements are always functions. This space can be complex, real, or even quaternion. However, in machine learning applications, these spaces are almost always real. Kernels along with RKHSs form a central theme for support vector machines. Additionally, kernel methods are employed in statistical machine learning algorithms. In this case, they employ a feature map Φ that is often used in converting data into higher or infinite dimensions. Higher dimensions often enable simpler information retrieval and classification. Among the reasons for this is that data that cannot be separated in a linear manner may become linearly separable in a higher dimension. A typical contrived example illustrating the use of higher dimensions to linearize data is given next. See also Fig. 1.4A and B. The former illustration provides data in two dimensions. This is before the feature map is used. The latter diagram illustrates what happens after the feature map is utilized.

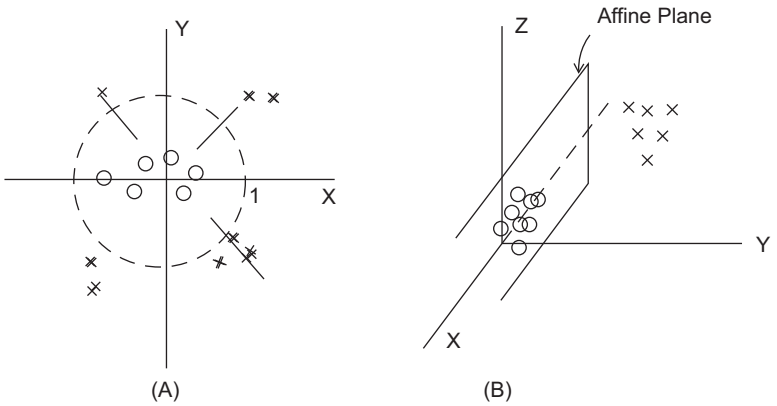


FIGURE 1.4 (A) Original data in R^2 and (B) feature mapped data in R^3.

Example 1.9:

Refer to Fig. 1.4A. That is, in the left portion of this diagram are data points denoted by o. These points are within the unit circle and are close to either the x axis or the y axis in R^2. The other data points in this diagram lie on or outside of the unit circle, and each point is

denoted by $\times$. Moreover, all these points have their x coordinate about equal to their y coordinate in absolute value. Consider the feature map Φ: $R^2 \rightarrow R^3$, where $\Phi((x\ y)') = (x^2\ xy\ y^2)' = (X\ Y\ Z)'$. Note that the resulting vector in three dimensions has its second tuple, that is, the Y tuple, equal to the product of the tuples in R^2. Then, in Fig. 1.4B, it is seen that the feature map transformed all the o points in an area near the plane $Y = 0$. Concurrently, it transformed all the $\times$ points at least one unit away from the plane $Y = 0$. It follows that affine planes close to, but also parallel to $Y = 0$, would separate the o points from the $\times$ points.#

Kernel methods are also useful in establishing independence and conditional dependence of random variables as well as separating signals from noise (Ravi and Kumar, 2013). An in-depth treatment of RKHS is provided in sections given in a later chapter.

Before the kernel method is described, another motivating example will be illustrated. It uses homogenous monomials of degree d, similar to the aforementioned example where $d = 2$. More generally, monomials are formed from a product of all possible tuples within a single vector v or w located in a low dimension Hilbert space, H_L. Each of these monomials is then used as the tuples for vectors in a higher dimensional Hilbert space, H_H. The mapping that performs this operation is Φ and is again called the feature map, Φ: $H_L \rightarrow H_H$. The kernel K(v,w) is given by the inner product of feature maps in the higher dimensional Hilbert space, $K(v,w) = < \Phi(v).\ \Phi(w) >$.

Example 1.10:

Consider the carrier set $H_L = R^2$ with the vectors, $v = (v_1\ v_2)' = (2\ 3)'$, and $w = (w_1\ w_2)' = (4\ -1)'$. Let the feature map Φ: $R^2 \rightarrow H_H = R^4$, where $\Phi(v) = \left(v_1^2\ \ v_2^2\ \ v_1v_2\ \ v_2v_1\right)' = (4\ 9\ 6\ 6)'$. Note that all possible monomials are formed, order matters, and all possibilities should be utilized. Similarly, operating on w, $\Phi(w) = \left(w_1^2\ \ w_2^2\ \ w_1w_2\ \ w_2w_1\right)' = (16\ \ 1\ -4\ -4)'$. Here, all possible monomials of order two became the tuples of vectors in R^4. Then, taking the inner product in R^4 of $\Phi(v)$ and $\Phi(w)$, $K(v,w) = <\Phi(v),\ \ \Phi(w)> = <\left(v_1^2\ \ v_2^2\ \ v_1v_2\ \ v_2v_1\right)', \left(w_1^2\ w_2^2\ \ w_1w_2\ \ w_2w_1\right)' > = <(4\ 9\ 6\ 6)', (16\ \ 1\ \ -4\ \ -4)' > = v_1^2w_1^2 + v_2^2w_2^2 + v_1v_2w_1w_2 + v_1v_2w_1w_2 = 25 = \left(<(v_1\ v_2)',\ (w_1\ w_2)'>\right)^2 = \left(<(2\ 3)', (4\ -1)'>\right)^2$.#

Thus it is seen that the inner product in the higher dimensional space is the same as the inner product in the lower dimensional space raised to the power $d = 2$. This is the crux of the matter; whenever the inner product in the higher dimension is desired, there is no need to utilize the feature maps. The result is obtained in the lower dimension simply by raising the inner product there to the power d, thus determining the higher dimension inner product.

More generally, the kernel method maps the dot product of two vectors in a Hilbert space $H_L = R^n$ into the dot product of two associated vectors in another Hilbert space $H_H = R_m$. When $n = m$, a linear kernel arises. In this case, $K(v,w) = <v,\ w>$, and for a in R, $<a\ \ v+u,\ \ w> = a\ \ <v,\ \ w> + <u,\ \ w>$. Similarly, $<v,\ a\ \ w+u> = a\ \ <v,\ \ w> + <v,\ u>$. Because it is linear, the kernel is $K(v,w) = v'w$ and the feature map $\Phi(v) = v$. The Gram matrix for this situation is an n by n matrix whose entries consist of inner products of vectors from R^n, specifically for $\{v_1, \ldots, v_n\}$ in R^n; then entry G_{ij} in G equals $<v_i, v_j>$.

Example 1.11:

Again consider the carrier set $H_L = R^2$ with the vectors, $v = (v_1\ v_2)' = (2\ 3)'$, and $w = (w_1\ w_2)' = (4\ -1)$. Assume that $H_L = H_H$ and that the feature map Φ is the identity; then the Gram matrix for v and w is given in general and again using the actual tuple values for v and w, G =

$$\begin{vmatrix} \langle v, v\rangle & \langle v, w\rangle \\ \langle w, v\rangle & \langle w, w\rangle \end{vmatrix} = \begin{vmatrix} 13 & 5 \\ 5 & 17 \end{vmatrix}.\#$$

When m is greater than n, $m = n^d$; then the dot product in R^n, raised to the positive integer power d, will equal the dot product in R^m. This is called the polynomial kernel method or the kernel trick. Here, vectors v and w in the domain space R^n are $v = (v_1\ v_2 \ldots v_n)'$ and $w = (w_1\ w_2 \ldots w_n)'$. The mapping function Φ: $R^n \rightarrow R^m$ takes all n tuples of a vector, for instance, v in the domain space, and forms all possible homogeneous monomials of degree d. When forming these monomials, tuples within the domain vector are to be repeated in different orders. These distinct monomials then become the tuples of associated vectors in R^m; that is, monomials formed from tuples of v become tuples in $\Phi(v)$. As far as the tuples are concerned, this is a nonlinear mapping. If this process is carried out, then $m = n^d$. Moreover, the kernel k is such that $k(v,w) = <\Phi(v), \Phi(w)> = (<v, w>)^d$. This result can be seen by taking the inner product in the higher dimensional space: $\langle \Phi(v),\ \Phi(w)\rangle = \sum_{j1} \cdots \sum_{jd} v_{j1} \cdots v_{jd} \cdot w_{j1} \cdots w_{jd,} = \sum_{j1} v_{j1} \cdot w_{j1} \cdots \sum_{jd} v_{jd} \cdot w_{jd,} = [\sum_j v_j \cdot w_j]^d = \langle v, w\rangle^d$.

Example 1.12:

In this example, let $n = 2$ and $d = 3$. For $v = (v_1\ \ v_2)'$ in R^2, $\Phi(v) = \left(v_1^3\ \ v_2^3\ \ v_1^2 v_2\ \ v_1 v_2^2\ \ v_1 v_2^2\ \ v_2 v_1^2\ \ v_2 v_1^2\ \ v_1 v_2 v_1\ \ v_2 v_1 v_2\right)'$ is a vector in R^8. If $w = (w_1\ w_2)'$, then homogeneous polynomials of degree three could fill the eight tuples of $\Phi(w)$, just as it did $\Phi(v)$. Assume that this is done, then forming: $\left\langle \Phi(v),\ \Phi(w)\right\rangle = \sum_{j1} \cdots \sum_{jd} v_{j1} \cdots v_{jd} \cdot w_{j1} \cdots w_{jd} = \sum_{j1} \sum_{j2} \sum_{j3} v_{j1} v_{j2} v_{j3} w_{j1} w_{j2} w_{j3} = \sum_{j1} v_{j1} w_{j1} \sum_{j2} v_{j2} w_{j2} \sum_{j3} v_{j3} w_{j3} = [\sum_{j=1}^{2} v_j \cdot w_j]^3 = \langle v, w\rangle^3 = [v_1 w_1 + v_2 w_2]^3 = v_1^3 w_1^3 + v_2^3 w_2^3 + 3 v_1^2 v_2 w_1^2 w_2 + 3 v_1 v_2^2 w_1 w_2^2.\#$

The benefit of using the feature map Φ is not using it. Dimension n data can be employed in a Hilbert space of dimension $m = n^d$ without actually transforming the data to this dimension. This is true whenever the inner product of the higher dimensional data is all that is desired. Here, the only calculation needed is simply to find $<v, w>^d$. Applications of the feature map and kernel methods will be provided in a later chapter, along with several feature maps and various kernels.

1.1.11 R-Modules

Again, to take advantage of the in-depth high-level definition in the MSA, a vector space substructure will be described. It is called unit R-modules over a commutative ring R with unity. The sorts are exactly as in the MSA vector space and field descriptions. So the sorts are again SCALAR and VECTOR. The same is true for the signature sets, except

that the partial operational name INV does not exist in a commutative unital ring, nor does it exist for a ring. Accordingly, in a polyadic graph as in Fig. 1.1, the resulting graph would be modified. In this case, the dotted arrow for INV would be removed entirely.

The actual signature sets for R-modules are as in a vector space structure. Accordingly, the arity sequence for an R-modules is again (1, 1, 2(1, 1)). The signature sets are as follows:

$$\begin{aligned}&\text{Binary operation } \{V-ADD\}, V-ADD \text{ maps VECTOR} \times \text{VECTOR} \to \text{VECTOR}\\ &\qquad\qquad\qquad\quad \{S-MULT\}, S-MULT \text{ maps SCALAR} \times \text{VECTOR} \to \text{VECTOR}\\ &\text{Unary operation } \{V-MINUS\}, V-MINUS \text{ maps VECTOR} \to \text{VECTOR}\\ &\text{Zero}-\text{ary operation } \{V-ZERO\}, V-ZERO = 0.\end{aligned}$$

Referring back to the equational identities for a vector space structure: (1)–(8), they all hold for unit R-modules.

Recall that constraints (1)–(4) imply that the structure being described, that is, a unit R-modulus, is also an abelian group under vector addition. Moreover, sometimes, the commutative property of a ring is left out in the definition of a module. If this is the case, the constraints (5) – (8) are adjusted accordingly. For instance, constraint (5) might be written as only: $1 \cdot v = v$. The 1, is a one-sided identity. Additionally, in this case, care must be taken in defining S-MULT. It can only be defined, taking the order of operations into account. In the MSA polyadic diagram, slashes are used marking the tails of polyadic arrow. In this type of diagram, a slash (/) is used for the first operand and double slash (//) is used for the second argument of S-MULT. The symbology is important in describing left or right modules. When the ring is a field, all this does not matter.

Example 1.13:

Every abelian group G can be considered as an unit R-module over the commutative ring with identity, namely the integer Z. In this case, SCALAR has a carrier set consisting of all the integers. The corresponding signature sets are as in the reals, except there is no partial inversion operator name. VECTOR has the carrier set G of all group elements. Scalar multiplication is as in a vector space:

S-MULT maps SCALAR x VECTOR → VECTOR,

Specifically, S-MULT (n; g) = g + g + … + g, n times where $n > 0$. For $n = -1$, S-MULT (n; g) = −g, for $n = 0$, S-MULT (n; g) = 0.

Also, the minus operation is as usual:

V-MINUS maps VECTOR → VECTOR,

However, it is defined using the scalar multiplication operation, namely V-MINUS(g) = S-MULT(−1; g) = −g.

The zero-ary operation is an element of the R-module, and as with other operations, it is defined using scalar multiplication:

So, V-ZERO = S-MULT (0; g) = 0.

Finally, the addition operation is as before in a vector space:

V-ADD maps VECTOR x VECTOR → VECTOR,

In particular, if h = n · g and k = m · g for n, m in Z and g in G. Then V-ADD(h, k) = S-MULT (n + m, g) = (n + m) · g = n · g + m · g.#

Of importance in quantum disciplines is when the modules are over or not over a division ring, for instance, the quaternions. When a division ring is used, there will always exist a Hamel basis. When the modulus is not over a division ring, there may or may not exist a basis. For instance, there exists a Hamel basis for vector fields over R^2, but there is no basis for vector fields on the two spheres (Eisenberg and Robert, 1979). The existence of a Hamel basis is important in quantum disciplines since infinite dimensional vector spaces may not have a norm and therefore a Schauder basis cannot exist. All this will be seen subsequently after the tensor product of Hilbert spaces is described.

References

Birkhoff, G., Lipson, B., 1970. Hetrogeneous algebras. J. Comb. Theory 8 (1), 115–133.

Eisenberg, M., Robert, G., 1979. A proof of the Hairy Ball theorem. Am. Math. Monthly 86 (7).

Feynman, R., 1986. Quantum mechanical computers. Found. Phys. 16 (6), 507–531. Springer Science and Business Media, LLC.

Gelfand, I., Naimark, M., 1943. On the imbedding of normed rings into the ring of operators on a Hilbert space. Google Books 3–20.

Giardina, C., 1991. Parallel Digital Signal Processing, A Unified Signal. Algebra Approach. Regency Publishing.

Goguen, J., Meseguer, J., 1986. Remarks on remarks on many-sorted algebras with possible empty carrier sets. Bull. EATCS 30, 66–73.

Goguen, J., Thetcler, J., 1973. A Junction between Computer Science and Category Theory I, IBM Research Report RC4526,

Gratzer, G., 1969. Universal Algebra. Van Nostrand Reinhold, New York.

Halmos, P., 1957. Introduction to Hilbert Space. Published by Chelsea Publishing Company.

Halmos, P., 1958. Finite Dimensional Vector Spaces. Van Nostrand Co.

Ravi, B., Kumar, T., 2013. Speech enhancement using kernel and normalized kernel affine projection algorithm. SIPIJ 4 (4).

Ribeiro, A., et al., 2020. Automatic diagnosis of the 12-lead ECG using deep learning neural network. Nat. Commun. 11, Article number 1760.

Segal, I., 1947. Irreducible representations of operator algebras. Bull. Am. Math. Soc. 53, 73–88. Available from: https://doi.org/10.1090/s0002-9904-1947-08742.-5.

This page intentionally left blank

CHAPTER

2

Basics of deep learning

2.1 Machine learning and data mining

Machine learning and data mining are global concepts utilizing databases with the goal of finding relationships, associations, and structures such as patterns within data (Wittick, 2014). What is actually learnt are algorithms and functions, or operations with appropriate parameter settings programmed within the learning machine. The underlying philosophy is to develop these procedures with as little human intervention as possible. The objective is to employ the learning process to classify, categorize, detect, or estimate characteristics pertaining to future data. Classification-type applications might be binary or have higher arity. In any case, the common problem to be solved is to distinguish and partition data into sets where the elements have common attributes. The estimation process might involve smoothing, filtering, or predicting. Smoothing refers to revisiting past data and making the best estimate of what happened in the past. Filtering refers to using data to determine the best estimates of the present. For instance, autonomous vehicles utilize sensor data as inputs to a learning machine. It filters the data and subsequently sends signals to control surfaces. This application might involve navigation, guidance, or some other control such as emergency alertness.

In general, future data is related to but distinct from the training data set. A training set always exists; however, the extent to which it is employed is summarized as supervised versus unsupervised learning. Supervised learning utilizes the training data in refining and determining parameters in training algorithms. A common attribute of supervised learning is labels attached to the data. These labels often partition the data by predetermined features. The two principal methods for machine learning that utilize supervised learning are neural networks and support vector machines. They are used both in detection and estimation. Support vector machines were mentioned in Section 1.10 where an introduction to kernel functions was given. These machines are described in detail in a later chapter. Neural networks that are often called deep learning machines are the main subject of this chapter. Convolutional neural networks (CNNs) are also briefly introduced in this chapter. They are described and illustrated with imaging techniques in a later chapter. Other types of neural networks exist. In particular, several types of recurrent neural nets (RNNs) are illustrated in a later section.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00026-6

Unsupervised learning is a technique wherein algorithms autonomously determine structure from the data. These techniques are numerous in machine learning. Unsupervised learning always employs domain knowledge and an overwhelming number of heuristic parameter settings. They range from elementary methods for clustering such as k-means or k-median techniques to other computationally intense source coding eigenvalue methods. Included with the eigenvalue formulations are the principal component transformation (PCT) and the singular value decomposition (SVD). These techniques along with quantum versions are also the topic of a later chapter. A simple procedure of unsupervised learning is illustrated later using k-means. A quantum version of this algorithm for data mining purposes is provided in a later chapter (Kavitha and Kaulgud, 2022).

The k-means methodology is a clustering technique. When applied to a finite ordered set X of vectors in R^n the procedure forms a partition of X. The k-medians method is almost identical wherein the median value of the data is employed instead of the mean or average. The number of subsets, m, $m > 0$, within the partition, is a heuristic, known from domain knowledge and must be specified a priori. Each ordered subset S_i, $i = 1, 2, \ldots, m$, is nonempty and involves a subset leader L_i. This value is the average value of all the points within the subset. The value L_i itself usually is not an element within S_i. To start the procedure, this average is calculated after initial conditions are employed. These initial conditions involve preliminary guesses to determine which elements of X are members of the subsets creating the partition.

After this, the recursive steps begin.

The recursive step starts with the actual average values being calculated for each subset. Specifically, the mean value L_i is found for each subset S_i. Note that each of the L_i is in general a vector with n tuples, since the average is found and recorded in each tuple. Then the one norm, distance d_i, is found between all elements in X and every subset leader L_i. Accordingly, for each located vector v in X, the distance calculation $\|v - L_i\|$ is performed for every $i = 1, 2, \ldots, m$. This calculation is also performed for every v in X, $v = (v_1\ v_2\ \ldots\ v_n)'$, $\|v - L_i\|_1 = |v_1 - L_{i1}| + |v_2 - L_{i2}| + \ldots + |v_n - L_{in}|$. The value obtained from this calculation is entered by vector location into d_i. Every d_i has the same number of entries, namely the cardinality of X. After the one norm calculations, a tuple-by-tuple comparison is performed among all the distance vectors d_i Locations of minimum distance are found, and from this, a clustering or regrouping is performed using the elements in X and the same value m. Each, possibly new, subset is created by using the elements closest to a subset leader. These steps are repeated over and over again until the stopping condition is employed. The stopping condition is when the partition does not change after a whole recursive step is performed.

Example 2.1:

For instance, X is a subset of R^1, where $X = (1, 2, 3, 4, 7, 8, 9)$. This set is written this way since data can be repeated. To begin, say $m = 2$, that is, only two sets will constitute the partition. The initial guess is the partition $S_1 = (1, 2, 3)$, and $T_1 = (4, 7, 8, 9)$. The next, iteration begins by calculating the subset leaders that are averages for these predetermined groupings. These values are real scalars because set X consists of real-valued elements. These average values are, respectively, $L_{S_1} = 2$, and $L_{T_1} = 7$. Next, the one norm distance

from each point within X to both subset leaders must be found. In this case, the one norm is just the absolute value. These distances are given as ordered tuples $d_{S_1} = (1, 0, 1, 2, 5, 6, 7)$, and $d_{T_1} = (6, 5, 4, 3, 0, 1, 2)$. Note that this is exactly the same ordering as the data in X. Tuple-by-tuple comparison to find the minimum is performed between d_{S_1} and d_{T_1}. So, to begin, since the number one in d_{S_1} is smaller than the number six in d_{T_1}, cluster S_1 will employ the first element.

Additionally, the 2nd, 3rd, and 4th tuples of d_{S_1} are also the smallest. However, the other tuples in d_{T_1} are the smallest. Now using this information, reclustering begins by creating a new partition still using $m = 2$. Assign points within a cluster that are closest to the subset leader. The new partition is $S_2 = (1, 2, 3, 4)$ and $T_2 = (7, 8, 9)$, As can be seen, this new clustering differs from the previous partitioning.

Accordingly, another recursive step in the procedure must be performed. So again, the corresponding leaders, that is, averages this time, are $L_{S_2} = 2.5$, and $L_{T_2} = 8$. Next, again compute the distance from all points in X to each subset leader. The results $d_{S_2} = (1.5, .5, .5, 1.5, 2.5, 3.5, 4.5)$, and $d_{T_2} = (7, 6, 5, 4, 1, 0, 1)$. Tuple-by-tuple comparison is performed similarly as before; this time between d_{S_2} and d_{T_2}. Here it is seen that the 1st, 2nd, 3rd, and 4th tuples of d_{S_2} are the smallest, but the other tuples in d_{T_2} are the smallest. Reclustering gives the same partition in this case. Therefore, since $S_3 = S_2 = (1, 2, 3, 4)$ and $T_3 = T_2 = (7, 8, 9)$, the stopping condition is utilized. The result is the k-mean partition.#

2.2 Deep learning

Deep learning is a major subset of machine learning wherein information is distributed and acted upon in parallel. Usually neural nets are employed. They consist of layers or columns of processors each designated to different tasks based on distinct data representations. Data entering the first layer, that is, the first column of nodes, may appear somewhat random. Algorithms in the nodes of the first layer often determine important features and transmit these features to nodes in the second layer. A pipeline-type architecture is frequently formed. Here, data is transmitted to subsequent layers usually creating higher conceptual information content. The last layer, the output layer, yields extremely high conceptual information such as facial recognition. Deep refers to the fact that many layers or columns in the neural net exist. Often there is a large mix of layers such as those used in CNNs. However, layers near the output usually consist of nodes employing affine functions along with activation functions. These nodes are trained in the conventional backpropagation gradient method. Supervised and semisupervised learning are employed in the deep learning process.

2.3 Deep learning and relationship to quantum

For over a decade, deep learning neural nets and artificial neural nets have been at the forefront of research and application. In particular, CNNs are heavily used in 2D or 3D imaging and machine vision. As previously mentioned, they are also employed in

one-dimensional signal applications (Wu, 2021). More recently, CNNs are being used in conjunction with or aiding quantum computers and quantum simulators. This research usually is directed at solving quantum many-body problems or quantum entanglement issues (Carlo and Troyer, 2022). The current basic goal is to build quantum versions of CNN (QCNN) (Choi and Kim, 2020). This entails the merging of quantum technologies with machine learning and neural net (NN) techniques. In particular, the NN affine structure for nodes is largely replaced by quantum entanglement. The MSA will help the development of applications involving both machine learning and QCNN.

2.4 Affine transformations for nodes within neural net

One of the simplest types of deep learning NN is the multilayer perceptron, which consists of arrays of nodes. It has been utilized in the classification of linearly separable patterns. Keeping all things mathematical, these arrays are described in the text, and in particular, in the next few sections by using a rectangular matrix organization for all the nodes. Neural networks utilize, at a local level, affine transformations. Perceptrons, neurons, or nodes within NN most often contain affine processing devices. Processing units, that is, each node, involves a column vector v in R^n arriving as input or coming from other nodes. The calculation within a node is of the form $u = W\ v + b$, where W is 1 by n real-valued matrix of weights. For $b = 0$, this is a linear transformation; otherwise, in all cases it is an affine transformation. The real-valued scalar b causes a translation for the weighted value Wv and is called the bias value. These weights and bias terms can be considered to be parameters that are adjusted to create some type of optimization. Once the value u is calculated using the affine transformation, this quantity usually undergoes a nonlinear operation T, and the results are sent to other specified nodes in the network. More details are provided in subsequent sections.

2.5 Global structure of neural net

As mentioned earlier, in deep learning neural nets affine transforms are used in each node or neuron, along with a nonlinear operation. The nodes are labeled N_{ij} and should be thought to be organized as in an n by m matrix N, $n > 0$, $m > 0$. Therefore, i denotes the row and j denotes the column for node N*ij*, throughout the text. To conform to the usual NN protocol, arbitrarily shaped arrays can be constructed from the matrix-type structure mentioned earlier by vacating specific nodes. In any case, corresponding to each connected node is the calculated value u_{ij}; at that node, the nonvacated, nonzero nodes are of utmost importance in this situation.

The quantity n is referred to as the depth, that is, the number of rows. Also, m is the number of layers or columns within the overall matrix structure. As previously mentioned, when the term deep learning is used, this often refers to a neural net with a large number of layers. Each column vector or layer is labeled by N_k, $k = 1, 2, \ldots, m$. Hidden layers, if they exist, are denoted by all labels except N_1 and N_m. Nodes in layer k are often connected to numerous nodes in layer $k + 1$, $k = 1$ to $m - 1$. Once the final layer N_m

outputs its result, this quantity is compared in some way to the input. Accordingly, a measure of the performance is created at this point. This will involve an objective or cost function. The correspondence might be an exact comparison, or it might involve important features concerning the input and final output. Actual implementations with examples are given in Sections 2.7 and 2.8.

The input to this device can be thought to be an nx1 column vector v. The output as previously mentioned is assumed here to be the last m^{th} column, N_m of N. The purpose of this device is to input an unknown vector v and then categorize, recognize, or determine this vector v by concluding that some function of N_m approximates v. This is performed utilizing some metric or closeness evaluation. The affine functions within the nodes N_{ij} involve weights w_{ij} and biases b_{ij}, $i = 1 \text{—} > n$, $j = 1 \rightarrow m$. The biases are real numbers, and the weights are real-valued row vectors. Their length is equal to the number of inputs coming into that specific node. As previously mentioned, inputs to any node are represented as tuples within a column vector. In the present section, these weights and biases are assumed known. However, in usual NN applications, values for these quantities are found in an iterative fashion.

The iterative procedure involves the optimization of an objective function to incrementally update the weight and bias parameters. The method usually is performed by backpropagation in a supervised training environment. Here, the predicted outputs of the NN are compared with the desired outputs using a loss or objective function. Subsequently, this function is optimized in an attempt to find a minimum. The procedure usually relies on gradient descent for the optimization. Accordingly, in practice the objective function and all activation functions must be continuously differentiable. After numerous training cycles and iterations, the network weights and biases might converge. When it does, it is at this point the NN is ready for use on actual data or more often on a test data set. The entire method is described and illustrated in the following sections of this chapter.

For the first column or layer N_1, each node N_{i1} has values of the form $u_{i1} = T(w_{i1}\ v_i + b_{i1})$, $i = 1 \text{—} > n$, where T is some nonlinear function called the activation function. Just to make a simple illustration, T might be a maximum, that is, $u_{i1} = \max(w_{i1}\ v_i + b_{i1}, 0)$. For the second column N_2, the nodes operate similar to those in N_1; here $u_{i2} = T(w_{i2}(\bullet) + b_{i2})$. This time, the weight w_{i2} multiplies whatever inputs it obtains from connecting nodes in column N_1. Total connections are not always needed; some, or even a few, nodes in column N_1 will connect to a specific node in N_2. This process continues usually with the same nonlinearity until outputs from $N_{m\text{-}1}$ enter the last column N_m. In this last column, often a different nonlinear function F might be employed, but the argument is still an affine function. So, at node N_{im}, $u_{im} = F(w_{im}(\bullet) + b_{im})$. The weights w_{im}, that is, these row vectors, multiply tuple by tuple all the inputs to N_{im} and add on b_{im}. After this multiplication and addition, the activation function F is calculated. Again, only connecting outputs from assigned nodes in column N_{m-1} enter nodes N_{im}, that is, the network need not be fully connected.

As mentioned previously, backpropagation algorithms are used to adjust assigned parameters within the affine transformations. Values for parameter updates are obtained using a gradient-type algorithm to maximize/minimize the chosen objective function. The procedure is performed in an exhaustive iterative fashion. Over and over until the NN completely learns, the output error is minimized. Usually, controlled test data is utilized for system validation. An in-depth description of the procedure begins in Section 2.7.

The next example illustrates a two-by-two NN structure. It is presented to illustrate some basic NN concepts such as affine transformations, activation functions, as well as the array structure and connections between nodes. No learning is illustrated in this example; moreover, the activation function utilized is not only nondifferentiable; it is not even continuous. However, this specific application shows that continuous functions can be generated using NN. By itself, this example provides a simple square pulse of height h, $h > 0$, between real values a and b, $a < b$, and is zero elsewhere. Linear combinations of such pulses provide a staircase-type function that can approximate any continuous function on the real line. Again, in this example, all parameters are fixed; there is no learning for the NN (Nielsen and Chuang, 2000).

Example 2.2:

Any continuous function f: $R \to R$ can be approximated by step functions. These functions are dense in the set of all real continuous functions, C(R). The two-by-two, $n = m = 2$, the neural net will provide a single step of height h with location in the interval (a, b] and zero elsewhere. In the first layer N_1, there exist affine transformations. They are followed by activation functions, T. In short, the output from each node in the first layer is $u_{i1} = 1$ for $(w_{i1}\ v_i + b_{i1}) > 0$ and $u_{i1} = 0$ otherwise, $i = 1, 2$. The affine functions and activation functions producing these results are given next. All double subscript labels are exactly as in an n by m matrix, $n, m > 1$.

The first layer N_1 has $w_{11} = 1$ and $b_{11} = -a$. Thus, $u_{11} = T(w_{11}x + b_{11}) = T(x - a) = \text{sgn}[\max(x - a, 0)]$, and so for x larger than a, this yields one; otherwise it yields zero. Here sgn(.) is the sign function; it is one for positive input values and zero otherwise. Similarly, $w_{21} = 1$ and $b_{21} = -b$. $T(w_{21}x + b_{21}) = T(x - b)$ yielding 1 if $x > b$. In the second layer, the activation function $F = I$; it is the identity function in this case. Also, in the second layer in N_{12}, the bias term is zero. In the second layer, there is a matrix weighting; it is a one-by-two row vector consisting of the pulse height, specifically, $w_{12} = (h\ -h)$. The row vector is employed because there are two inputs, to N_{12}. In vector form, the input to this node is the column vector $(y_1\ y_2)'$. In the present NN application, the node N_{22} is not used. Refer to Fig. 2.1.

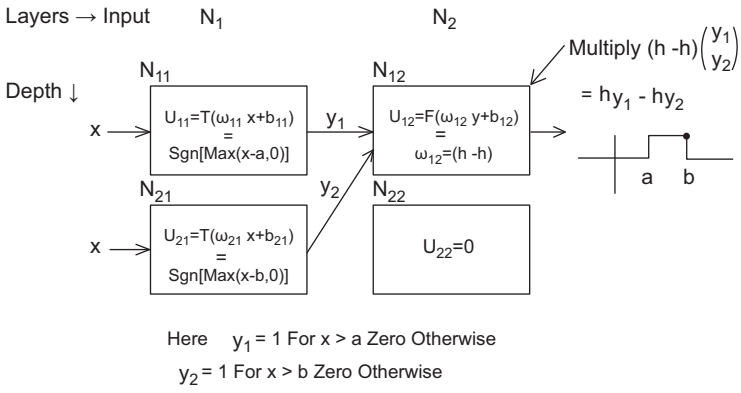


FIGURE 2.1 Matrix structure for NN square wave pulse creation. *NN*, Neural net.

For the first layer, the input vector can be thought to be the column vector, $v = (x\ x)'$. In practice, values of real numbers x, in very small increments, can be inputted simultaneously for both tuples of v. By connecting the output, $y_1 = u_{11}$ of node 11 as input to node 12, and the output of node 21, $y_2 = u_{21}$ also as the input of node 12 yields a rectangular pulse. This pulse is on the x-axis and is of height h with support approximately, (a,b]. Notice that Fig. 2.1 provides a block diagram. It is given in strict matrix form, illustrating the operations needed to create a square pulse. Each node is denoted by a rectangle. Each active node contains an affine function and an activation operation. Again, note that node N_{22} is not employed in creating the square wave pulse.#

Preparing for Section 2.7, where NN is described in more detail, particularly for the supervised learning aspect, a more pictorial presentation will be given. Here, it is convenient to symbolically portray the algebraic operations within a node in greater detail. The symbolic representation provided herein is inspired by Petri nets (Petri, 1962; Graff and Giardina, 2005). These nets are also called place and transition nets.

In the present case, internal operations within node N_{ij} are illustrated by using a triangle representing addition. This symbol is utilized for adding the scalar bias b_{ij} to the sum of weighted inputs as well as creating this sum. A single circle encloses each individual scalar-valued weight within the row vector w_{ij}. The circle symbolizes a multiplication operation. Again, the scalar weights within the row vector multiply tuple by tuple the column vector of input tuples for the node N_{ij}. Finally, a square is employed in identifying the activation function, T or F. In general, the activation functions are invariant during the learning process. However, the output from the activation function T or F usually does change because the input to these functions themselves changes. Refer to Fig. 2.2 and the example below, Example 2.3 to see an instance of this new notation. In the referenced diagram, a simple node N_{ij} is illustrated. Here, numerous inputs are multiplied by an encircled individual scalar weight. Each individual weight is a distinct tuple within the row vector w_{ij}. Subsequently, all the products are added together in the three-sided figure along with the bias b_{ij}. Lastly, a square indicates an activation function for that node. The previous example will be repeated, this time with the new visual symbolic operations within nodes of the NN.

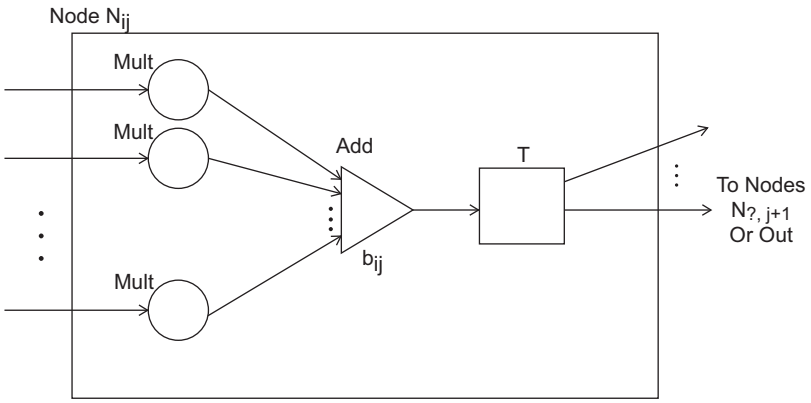


FIGURE 2.2 Symbolic schema for operations within NN nodes. *NN*, Neural net.

Example 2.3:

As in the previous example, a rectangular pulse function is created using the NN; this time, it is illustrated in Fig. 2.3. As mentioned, the internal structure of each node is portrayed with symbols. Nodes $N_{1,1}$ and $N_{2,1}$ are identical except that the biases are opposite in sign. Again, the activation functions in both nodes are $T(w_{j1}x + b_{j1}) = \text{sgn}\,[\max\,(x - b_{j1}, 0)]$. Note that in node $N_{1,2}$ the activation function is not utilized; thus the identity function is illustrated within the square. Also, in this node two inputs arrive. Accordingly, two distinct weights are utilized, indicative of heights h and $-h$. These are each enclosed within a circle. Together, these weights are entries or tuples within the row vector $w_{12} = (h \ -h)$. Finally, as before, node $N_{2,2}$ is not employed in this NN.#

2.6 Activation functions and cost functions for neural net

Conventional NNs have nodes utilizing affine functions followed by an activation function F, which is almost always nonlinear. Common attributes for an activation function are that when the output of the affine map is the input to F, then F often returns values in $[-1, 1]$ or $[0, 1]$. In the latter situation, the output is understood as a probability value. More importantly, when backpropagation is employed F must have a continuous derivative. When this is not the case, a noncontinuous derivative for F has the property that very small changes in the input can cause drastic deviations in the value of the output derivative. See the example below.

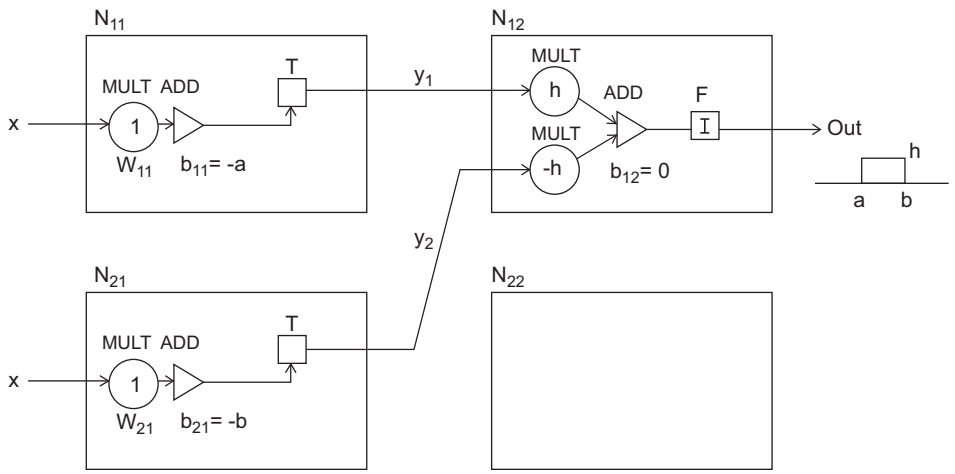


FIGURE 2.3 Symbolic calculations for NN square wave pulse creation. *NN*, Neural net.

Example 2.4:

The function f: $R \to R$, where $f(x) = x^2 \sin(1/x)$ for x nonzero, and $f(0) = 0$. This function is differentiable with $f'(x) = 2x \sin(1/x) - \cos(1/x)$ for x nonzero, and $f'(0) = 0$. So, for instance, if $x_1 = 0$ and $x_2 = 1/(1000\pi)$, then $f'\{x_1) = 0$, but $f'(x_2)$ is one.#

Activation function must often be chosen such that when used in backpropagation the gradient does not vanish or explode. When the gradient becomes too small, weights and biases more or less stay invariant. On the other hand, exploding gradients cause instability, which almost never results in learning. This is a large problem in RNNs due to the feedback; however, modified RNNs described in Section 2.11 were created to overcome some of these problems.

Several commonly used activation functions are listed below along with their range and their derivative, when the latter exists. These functions are defined on the real line,

Sigmoid: $T(x) = 1/(1 + e^{-x})$; range: (0, 1); derivative: $T'(x) = T(x)\ (1 - T(x))$
Hyperbolic Tangent: $T(x) = 2\ (1 - e^{-2x})^{-1} - 1$, for x not zero, and -1 at zero; range: $[-1, 1]$; derivative: $T'(x) = (1 - T^2(x))$
ReLU: $T(x) = 0$ for $x < 0$ and $T(x) = x$ otherwise; range: $[0, \infty)$; derivative: $T'(x) = 0$ for $x < 0$ and $T'(x) = 1$ for $x > 0$; $T'(0)$ does not exist.
Soft log: $T(x) = \log(1 + e^x)$; range: $(0, \infty)$; derivative: $T'(x) = 1/\ (1 + e^{-x})$
ArcTan: $T(x) = \tan^{-1}(x)$; range: $(-1, 1)$; derivative: $T'(x) = (1 + x^2)^{-1}$.

One of the most commonly used activation-type functions for multiclass classification is Softmax. It is almost always employed on outputs from the final layer of a NN. Accordingly, from this perspective, Softmax can be thought to be a pseudo node. It is of the type many inputs to many outputs. Therefore, outputs from all n nodes in the final layer, N_m, enter Softmax. Say that these outputs are $x_1, \ldots, x_n$; then

Softmax $(x_i) = (e^{x_i})/(\sum_j e^{x_j})$. Moreover, it is assumed that the pseudo node keeps track of the inputs sent from all final nodes, $N_{i,m}$, and sends as outputs the Softmax calculations in order. In this case, the output from any actual final node can itself be thought to be a probability. Because, after the final node the output value enter the pseudo node, it gets calculated by Softmax and takes on values in [0, 1]. And the sum of all the final node outputs when converted by Softmax is one.

Example 2.5:

Say that three final nodes exist in a NN. Each such node sends its output to the pseudo node Softmax. If the actual outputs of each final node are given in order by (2, 1, 0.5), these values enter the pseudo node. The Softmax operation is applied three times and provides in order the actual probability type vector that approximately equals (0.6, 0.25, 0.15). From this output, the final node one results in the correct classification.#

Various objective or cost functions are employed in machine learning. Most of the time, these functions are convex. In R^1, f is convex in an interval [a, b], which means that for x and y in this interval, $x < y$, and any λ in [0, 1], $f(\lambda x + (1 - \lambda)\ y)$ is less than or equal to $\lambda\ f(x) + (1 - \lambda)\ f(y)$. This means the chord for f in the interval [x, y] is larger or equal to the value of f in this interval. Convex objective functions are used since in this case local

minima are also the global minima. In later chapters, nonconvex objective functions are employed making use of simulated annealing and in particular quantum annealing techniques. Annealing-type operations often enable global minimization to occur (Laarhoven and Aarts, 1987).

Example 2.6:

Refer to Example 2.1, where the k-means procedure is illustrated using the one norm objective function. In this case, since data from R^1 was employed, the one norm is $\|v\|_1 = |v|$. For λ in [0, 1] and any x and y in R^1, $|\lambda x + (1-\lambda)|$ is less than or equal to $\lambda|x| + (1-\lambda)|y|$ by the triangle inequality. So this objective function is convex.#

Other important convex cost functions are quadratic, for instance, least squares and methods involving variance reduction such as square error.

2.7 Classification with a single-node neural net

Recall, from Section 2.5, that nodes N_{ij} within an NN have addresses exactly like entries within a matrix; the quantity (i, j) represents the ith row and jth column, respectively.

A circle is used, each enclosing an individual scalar-valued weight represented as a tuple within the row vector w_{ij}. Moreover, the circle itself symbolizes a multiplication operation. Also within the node, a triangle representing addition symbolizes adding the scalar bias b_{ij} to the sum of weighted inputs to node N_{ij}. Finally, a square is employed in identifying the activation function, T or F, and causes the activation function to activate and provide a scalar output. Section 2.6 provides a description of several activation functions. In this section, a single-node NN is employed. For this situation, subscripts won't be used; moreover, the input value v will be a real number in a specified interval. It will enter the node and then enter the circle-type figure containing the specified weight. Multiplication by the weight occurs here. The output from the circle diagram is labeled x and enters the triangle, wherein the bias value b is added to x. The output from the triangle is y, and this value enters the square-type figure. Here, the activation function F is evaluated, and $z = F(y)$ is the output value. This whole scenario is explained in Example 2.7 and is illustrated in Fig. 2.4.

Example 2.7:

The objective is to build an NN whose input could be considered to be in dollars, v in the interval $[0, \infty)$. The output z is to flag any input that is greater than 50,000; in this case, the output z is set to one; otherwise, it is set to zero. The single node NN performing this binary classification is illustrated in Fig. 2.4. In this case, whenever an input v enters, it gets multiplied by $w = 1$, and so $x = v$; this quantity is added with the bias $b = -50{,}000$, resulting in the value $y = v - 50{,}000$. This quantity is used as the input of the activation function, $F(y) = \text{sgn}\,[\max(y, 0)] = \text{sgn}\,[\max(v - 50{,}000, 0)]$. Thus, when $v > 50{,}000$, a one appears for z. Otherwise, the value $z = 0$ appears. This is the exact same activation function used in Examples 2.2 and 2.3.#

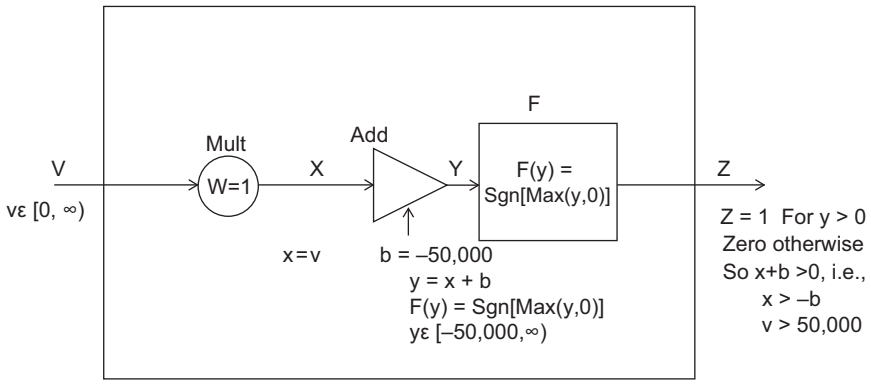


FIGURE 2.4 Classification using NN with noncontinuous activation. *NN*, Neural net.

The activation function employed in the earlier example is usually not used in NN applications. First of all, in this example, the max function is continuous, but not differentiable. Worse yet, the sgn function is not even continuous. As mentioned previously, NN applications normally need continuously differentiable functions. This is due to the backpropagation learning aspect, which is essential in all cases.

Since the activation function utilized in Example 2.7 is almost never used, the example will be performed again using the sigmoid function. In this case, the output z can be understood as a probability. Again, this is a classification problem; now the value of z will either be greater than 1/2 for the affirmative input, that is, when the input is greater than 50,000. If the input is less than or equal to 50,000, the value will be 1/2 or less.

Example 2.8:

Again, assume that the input is in dollars, v from the interval $[0, \infty)$. The output z gives a value $z > 1/2$ whenever any input is greater than 50,000; otherwise, it is set to 1/2 or less. The single node NN performing this binary classification is illustrated in Fig. 2.5. Once more, whenever an input v enters, it gets multiplied by $w = 1$, and so $x = v$; this quantity gets added with the bias, $b = -50{,}000$, resulting in the value $y = v - 50{,}000$. The resulting value is the input to the new activation function, $F(y) = 1/(1 + e^{-y}) = 1/(1 + e^{-(v\text{-}50000)})$. Thus, when $v > 50{,}000$, a number between $1/2^{+}$ and one appears for z. Otherwise, z is a number between zero and 1/2.#

2.8 Backpropagation for neural net learning

In this section, initially it is assumed that the structure of the full NN is defined, so all biases and weights are fully known. Moreover, this includes all activation functions T or F, as well as the chosen error criteria, C. Section 2.6 describes several well-known activation functions and error measures. The backpropagation procedure to train an NN is supervised learning in which an exhaustive amount of computation must be performed. It

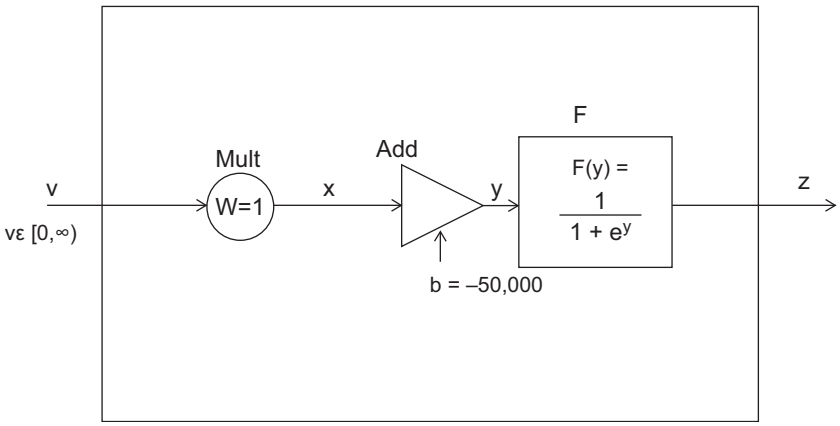


FIGURE 2.5 Classification using NN with sigmoid activation. *NN*, Neural net.

begins by using domain knowledge of the application so that educated guesses are provided for each weight and bias value within every node. Often, when domain knowledge is not available, weights are chosen randomly and biases are often set equal to zero. Also needed is the training data. One by one, each piece of training data x will enter and be executed or processed by the NN. Ultimately, an output z from the NN appears. This output is also called the predicted output. The value z is used in the cost or objective function C. It is then compared with the actual, true, or desired output, d. The desired value is known prior to starting the NN backpropagation process. The value d depends on the actual input v. Calculating the cost results in an error ϵ, which, when not zero, indicates the need for recalibration of weight and biases using backpropagation along with the gradient decent method.

For each node value to be updated, the chain rule is used in describing the changes in C. These changes ΔC are due to variations in weights ΔW or changes in bias, Δb values. Updated values are needed in order to apply the gradient algorithm. Each result from the chain rule becomes a value of the slope in the gradient descent algorithm. Then, the gradient algorithm is employed involving C and any weight or bias that is to be updated. In all cases, ΔW or $\Delta b = -\eta$ grad(C), where η is a positive number called the learning rate and is heuristically found. The minus sign is used since the gradient points in the direction of maximum change, but here the cost is to be minimized. The gradient descent begins with initial conditions using index $k = 0$. Then, for $k = 0, 1, 2, \ldots$, the algorithm computes: $x_{k+1} = x_k - \eta_k \, f'(x_k)$. As previously mentioned, the step size is η_k, and it is heuristically found.

After the aforementioned calculations, the whole procedure repeats; again, substituting in the new values of weights or biases and executing the NN again with these new values, and then finding new predicted outputs d, calculating the cost, and so on. A sequence of examples will be presented in order to illustrate the backpropagation and chain rule protocol. All these examples utilize the situation given in Example 2.8, where the actual solution is well known beforehand. In actual NN applications, this is never the case. Deliberate changes are made to this example, one by one, one at a time, to the weight or the bias in

the ensuing examples. By doing this, the backpropagation and the chain rule methodology become more understandable, and the path to the solution via backpropagation becomes more apparent. To repeat, several examples will follow wherein the exact solution is known. However, a deliberate wrong setting is made on the bias or weighting term, and the backpropagation and chain rule calculations will somewhat correct the wrong settings.

Example 2.9:

Refer to Example 2.8, where the input v is a value in the interval $[0, \infty)$. The output z should be a value $z > 1/2$ whenever any input is greater than 50,000; otherwise, it is set to 1/2 or less. But this time, the proposed output deliberately will not occur. In Fig. 2.5, the actual NN employed in this example is modified for this example as well as the next two examples. Observing the following figure, three distinct sets of conditions are indicated. Each set has a single deliberate error in either the weight or the bias value. These values are listed in order below the transmission from the circle to the triangle in the diagram. The values should be applied to this example, followed by Examples 2.10 and then 2.11 in that order.

In the present example, when an input v enters, the correct weight is kept, $w = 1$, and so $x = v$. However, this time let the bias be set to $b' = -51{,}000$. Fig. 2.6 illustrates the same NN as in the previous example, but for this problem a new value of bias is used; it is $b' = -51{,}000$, but the weight stays the same; it is $w = 1$. Using the new value of bias results in the value $y = v - 51{,}000$. So, any input value less than 51,000 will not be flagged with this NN structure. Accordingly, this can result in misclassification. For instance, when the value 50,999 enters, a value closer to zero than one is obtained as the predicted output. This is arrived at by inputting $y = v - 51{,}000 = -1$ into the activation function $z = F(y) = 1/(1 + e^{-y}) = 1/(1 + e) = 0.27$. So in this case, the actual observed output of the NN is much less than 1. However, the desired output is $d = 1$, since the input is $v > 50{,}000$. Employing the objective function C, gives $C = (d - z)^2 = 0.53 = \epsilon$. The value of this

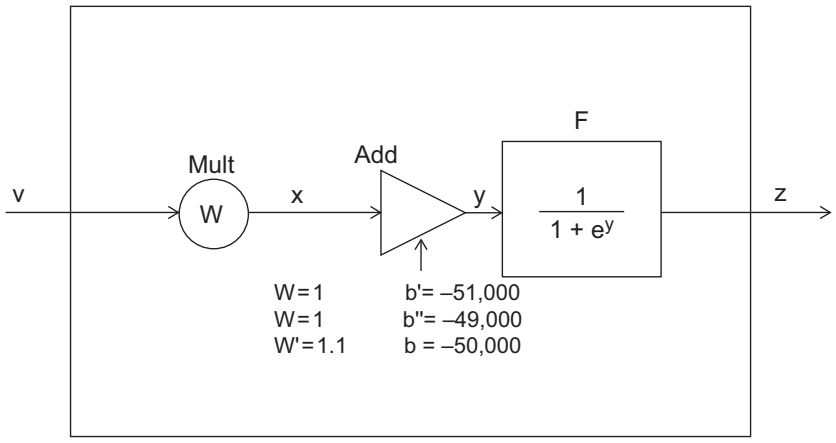


FIGURE 2.6 Bias and weight modifications for single node NN. *NN*, Neural net.

objective function is another indication of the actual error. However, after using backpropagation, the new value for the error will be less than 0.53.

Beginning with the cost function and using the chain rule, the objective is to update the bias such that the cost function decreases. So to begin, the change in the cost function due to a change in the bias is desired; it is dC/db. Use the chain rule, go backward, and refer to the Fig. 2.6, mentioned above. This change depends on how the cost function decreases with respect to a change in the output z. This output arises from the sigmoid function, specifically it is dC/dz. Remember that the output of the sigmoid function varies because its input y has changed; this is summarized as dz/dy. Finally, the value y can change because of a variation in the bias; thus dy/db symbolizes this change. Utilizing the chain rule gives the desired backpropagation formula, $dC/db = dC/dz\ dz/dy\ dy/db$. Substituting actual values gives $dC/dz = -2\ (d - z) = -1.46$ and $dz/dy = F'(y) = F(y)\ (1 - F(y)) = .27 \cdot 0.73 = 0.2$. Finally, $dy/db = 1$, and so multiplying all these together gives $dC/dz = -0.3$. The negative of this value will be utilized in the gradient algorithm. Using a learning rate η, say $\eta = 1000$, then adding 300 to the current bias $b' = -51{,}000$ provides the new value of bias $b'^* = -50{,}700$. This calculation was $b'^* = b' - \eta\ dC/dz$.

The aforementioned description presented the gradient descent algorithm in a nutshell. The parameter setting for the learning rate η was employed; since we knew the true answers ahead, we cheated. When the bias was set to the wrong value, that is, $b' = -51{,}000$, then the cost or loss function was seen to be $\epsilon = 0.53$, The important thing to notice is since the value of bias that was set in this example was too small, the algorithm gave a positive increase in bias change. If another iteration of backpropagation was performed, another small increase in bias would occur. Ultimately, the true value of bias, $b = -50{,}000$, should be obtained.#

The next example will again illustrate the backpropagation procedure. This time, the bias b″ is set to be too large a value. Here, it will be seen that the procedure will provide a decrease in the bias value.

Example 2.10:

Again, referring to Example 2.8, the input v is a value in the interval $[0, \infty)$. The output z should be a value $z > 1/2$ whenever any input is greater than 50,000; otherwise, it is set to 1/2 or less. Again, when an input v enters, the correct weight is kept, $w = 1$, and so $x = v$. Now, a value of bias greater than the optimal value will be used, so let $b'' = -49{,}000$, resulting in the value $y = v - 49{,}000$. Again see Fig. 2.6, for the actual NN employed in this example; here $w = 1$ and $b'' = -49{,}000$. So, any input value greater than 49,000 will be flagged, with a value greater than 1/2, even though it should not. This again results in misclassification. Here, for instance, when the value 49,001 enters, a value closer to one than zero is obtained as the predicted output. The calculation occurs by inputting $y = v - 49{,}000 = 1$ into the activation function, $z = F(y) = 1/(1 + e^{-y}) = 1/(1 + e^{-1}) = 0.73$. In the current case, the desired output is $d = 0$, since the input is $v < 50{,}000$. We use the objective function $C = (d - z)^2 = 0.533 = \epsilon$. This nonzero value also indicates an error. Accordingly, the use of the gradient algorithm along with back projection will be needed.

Again using the chain rule, exactly as in the previous example, $dC/db = dC/dz\ dz/dy\ dy/db$, where $dC/dz = -2\ (d - z) = 1.46$ and $dz/dy = F'(y) = F(y)\ (1 - F(y)) = 0.733 \cdot (1 - 0.73) = 0.196$.

Finally, $dy/db = 1$, and so multiplying all these together gives 0.29. Since the negative value of 0.29 is employed in the gradient algorithm, this automatically provides a negative increase in the bias. It is of paramount importance that the bias gets closer to the desired value. Since the true answer is known, set the learning rate $\eta = 1000$; then the new value of bias is given by $b''^* = b'' - \eta\, dC/dz = -49{,}000 - 290 = -49{,}290$.#

A final example will be given, this time keeping the bias at the correct value but slightly changing the weighting function value. In this case, a weighting function too large is employed. The backpropagation and gradient technique will indicate this error and correct it somewhat. However, in general, these algorithms must be employed over and over again to possibly obtain the optimal solution.

Example 2.11:

Again, referring to Example 2.8, the input v is a value in the interval $[0, \infty)$. The output z should be a value $z > 1/2$ whenever any input is greater than 50,000; otherwise, it is set to 1/2 or less. Similar to before, the proposed output will not occur, this time because the wrong weight w′ is utilized, but the correct bias is still employed. See Fig. 2.6 for the NN illustrating this example. When an input v enters, an incorrect weight, $w' = 1.1$, is used, and so $x = 1.1 \cdot v$. The bias is set to $b = -50{,}000$. Doing the calculations, any input value greater than 45,454 and less than 50,000 with this NN structure will result in misclassification. For instance, when the value 45,455 enters, a value closer to one is obtained as the predicted output. Note that $x = w \cdot v = 1.1 \cdot 45{,}454 = 50{,}001$. Furthermore, by inputting $y = x - 50{,}000 = 50{,}001 - 50{,}000 = 1$ into the activation function, $z = F(y) = 1/(1 + e^{-y}) = 1/(1 + e^{-1}) = 0.73$. In this case, the desired output is $d = 0$, since the input is $v < 50{,}000$. We use the objective function $C = (d - z)^2 = 0.53 = \epsilon$. This value also indicates the need for the gradient algorithm, with back projection given next.

By the chain rule, $dC/dw = dC/dz\ dz/dy\ dy/dw$, where $dC/dz = -2\ (d - z) = 1.46$ and $dz/dy = F'(y) = F(y)\ (1 - F(y)) = 0.73 \cdot (1 - 0.73) = .96$. Finally, the change in y due to the changes in weights is different from the change in y due to the changes in bias. In this case, the input is involved, $dy/dw = v = 45{,}454$. Lastly, multiplying all these together yields 63,709. However, the negative of this value is utilized in the gradient algorithm. Here, say that the learning rate is set to $\eta = .000001$. The new weight with $w''^* = 1.1 - 0.063709$, which is about 1.036, is a value closer to the true value of w.#

2.9 Many-sorted algebra description of affine space

Affine space consists of a nonempty set A with elements a and b along with a supporting vector space V; V is called the translation space for A. It is such that for the difference of any two points of A, there is a corresponding vector, v in V, that is, $a - b = v$. In particular, $a - a = 0$. There is a sought of inverse, write $b = v + a$. This equation is interpreted to mean that a is translated along the vector v to provide the point b. When $v = 0$, $b = a$, that is, there is no translation. In any case, consider the mapping, $+ : A \times V \to A$. So in this case, the vector space acts on the set of points in A. Vectors in V move elements in A; that is, vectors in V translate elements in A to new locations (Berger, 1964).

An affine space A is a principal homogenous space for the vector space V; that is, A is nonempty, and for a and b in A, there is a unique vector v in V such that a + v = b (Lang and Tate, 1958). The affine space has a free and transitive action of V on A. This property is also called a V-torsor; that is, the vector space V acts as a translation on A employing a left or right torsor with its additive group. In the MSA approach, there are actually three sorts: SCALAR, VECTOR, and ELEMENTS. However, in Fig. 2.7, only two sorts are shown; that is, SCALAR is omitted. As usual, VECTOR refers to a vector space, but the operator names solely within the signature sets in a vector space are not illustrated in this diagram. ELEMENTS refer to the nonempty affine space, which intuitively looks like a vector space without the zero. To make A a V-torsor, a single signature set is needed with a single binary operational name TRANS. It is such that

TRANS: A x V → A.

Additionally, TRANS must obey three equational constraints. To see these, let v, w, and 0 denote sort VECTOR and let a be an ELEMENT. Also use + for TRANS.

1) Null Translation: a + 0 = 0 + a = a
2) Associative: a + (v + w) = (a + v) + w
3) Isomorphic Map: A x V → A x A, (a, v) → (a, a + v)

An affine space A can be visualized as a line, plane, or hyperplane in a vector space V that does not go through the origin. For instance, consider $V = R^3$, elements u and v in A; the sum u + v is most often not in A. The same holds for a scalar multiple a · u, where a is in R. Any such affine space is parallel to a plane S going through the origin. This plane S is a subspace of V. Choosing any fixed element w in A, the difference u − w is in S, as well as v − w. This defines a one-to-one correspondence between A and S. From this correspondence, addition and scalar multiplication can be defined on A. So, using fixed w in A then V-ADD(u,v) in A, means u + v − 2w, is in S. Similarly, S-MULT(a; v) in A means that a v − a w, is in S.

Example 2.12:

In $V = R^3$, consider the subspace S defined by the set of points (x, y) that satisfy the equation 2 x + 3 y = 0. Let the affine space A be defined by the set of points on the plane: 2 x + 3 y = 6. Notice that this affine space can be multiplied by any nonzero scalar. For instance, multiplication of the last equation by 2 gives 4 x + 6 y = 12. Thus, there exist equivalence classes of affine spaces. Let w be the fixed point (0, 2) in A. To add u = (3, 0)

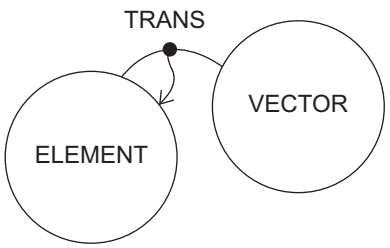


FIGURE 2.7 Polyadic graph of affine space.

and v = (6, −2), both in A, use u − w = (3, −2) in S and v − w = (6, −4) also in S. To add u and v, referring to the aforementioned formula, take a = b = 1; then u + v −1 w = (9, −2) − (0, 2) = (9, −4) is in A.#

Example 2.13:

Let the carrier sets for both SCALAR and VECTOR be the real number field, R. Also, see Example 10.5, where this structure with the usual conditions is a vector space. Even more is true; however, it is a Hilbert space. In this case, inner products can be formed. Vectors v in R can be thought to be arrows going horizontally right or left with their tail located at the origin. V-ZERO is the vector 0. So, for instance, if v = 7 and w = −3, then V-ADD(v, w) = u = 4. Note that dot or inner product, <v, w> = −21, and $\|w\|_2^2 = 9$ are scalars in R. Also, using the one norm, the length of w is $\|w\|_1 = 3$. Finally, consider the affine space with sort ELEMENT also being R. Then taking the point say $P_1 = 8$, TRANS (P_1; v) = P_2 = 15. The vector transported the point on the real line from location 8 to location 15.#

2.10 Overview of convolutional neural networks

Convolutional neural networks are the premier example of deep learning structures. Applications of CNNs are used for solving recognition as well as classification problems. However, CNNs are mostly employed for data having a locally correlated structure, and they are less useful for random-type data (LeCunn et al., 1989). Applications range from facial recognition to their use in self-driving vehicles employing three-dimensional data. A later chapter provides more details and examples of CNNs for these and other image-type applications. Here, mainly single-dimensional CNNs are briefly described. One-dimensional CNNs are applied to sound, time series, or data streams such as stock market prices.

Forward propagation is the first step in a CNN and will be described next. The specific architecture is application dependent, but it usually involves several layers of convolutional nodes. Different filters or kernels are applied in different layers. In each layer, convolution involving bound vectors from input signals or output signals from previous nodes is convolved with specified kernels. Sometimes a bias is added again creating an affine-type operation within a node. These layers are often stacked, but are applied in a feed-forward manner. To achieve better results, backpropagation is sometimes employed in modifying the filters or kernels used in the convolution, as well as biases whenever they are unemployed. Here, standard gradient techniques are utilized along with the back-propagation process.

After convolutional layers are performed, or even sometimes in between convolutional layers, a pooling layer is usually applied. This is a compression or source-coding operation also called downsampling. It condenses or summarizes the information from the previous layer. The pooling method is usually a simple averaging of a specified group of data, and this average is used instead of the group of data itself. Often max pooling is performed, wherein the largest value for a specified group of data is chosen to replace the group of

data. Additionally, a normalization is sometimes performed using truncating methods; for instance, ReLU is utilized. This operation is employed to make all values nonnegative. Stacked layers involve several convolutions, ReLU, or pooling layers all in some specified order. In every case, a reshaping operation is needed for two or higher dimensional applications. This operator converts unordered information into a strictly ordered vector of information to input into a conventional NN. Finally, dense layers, one or more fully connected NN layers, are applied. It is trained with backpropagation to determine optimal values in the affine transform within each node. This entails operations similar to those described in a previous section.

Hyperparameters are heuristically determined parameter settings needed for the operation of the NN. They are found from domain knowledge as well as trial and error or from transfer knowledge. For the overall architecture, hyperparameter decisions are the number of convolutional layers, the number of pooling layers, and the number of ReLU layers. Additionally, parameters are needed for the associated connecting deep NN, along with parameters for the number of features, types of kernels or filters, as well as their size and stride. The latter parameter deals with the number of zeros inserted in between data values in the pooling and convolution layers. Stride in the bound vector convolution process is mentioned again with examples in Section 3.2. Other hyperparameters are the types of layers as well as the number of each layer and the order of application.

2.11 Brief introduction to recurrent neural networks

Recurrent neural networks (RNNs) are useful for time series data and other sequential streams of input values. The main attribute of the RNNs is its recursive feedback feature. It enables data predictions or filtering, using past and present data. Applications include text classification, text generation, time series forecasting, financial applications, and medical sketching. RNNs can utilize data of different dimensions, however, they have their own problems. Including vanishing sigmoid activation functions, and exploding gradient property. So when backpropagation is utilized often, the method is troublesome. Whenever the NN weights are greater than one, the gradient explodes. On the other hand, when the weights are less than one, the gradient goes to zero. Vanishing and exploding gradients make training very difficult. A partial solution is clipping when the gradient gets too large and re-initializing different biases when the gradient gets too small.

A well-known solution to the exploding or vanishing gradient involves two modifications of the RNNs. The first is using long short-term memory (LSTM) machines (Hochreiter and Schmidhuber, 1997). The other is using gated recurrent units (GRUs). All three NN machines each consisting of a single node are illustrated in Fig. 2.8. Symbology utilized in this diagram is described in Section 2.5 and illustrated in Fig. 2.2. Every circle contains a weight, and the inputs to the circle are all multiplied together. A triangle adds together all its inputs, and an arrow pointing to the triangle denotes a bias, which is added to other inputs. A square indicates an activation function, and the input is the argument. The sigmoid in the diagrams is denoted by σ, and the hyperbolic tangent is denoted by T.

A single node of the RNN machine is illustrated in Fig. 2.8A. In this diagram, the feedback connection is clearly illustrated. In applications, these nodes are replicated, yielding an RNN

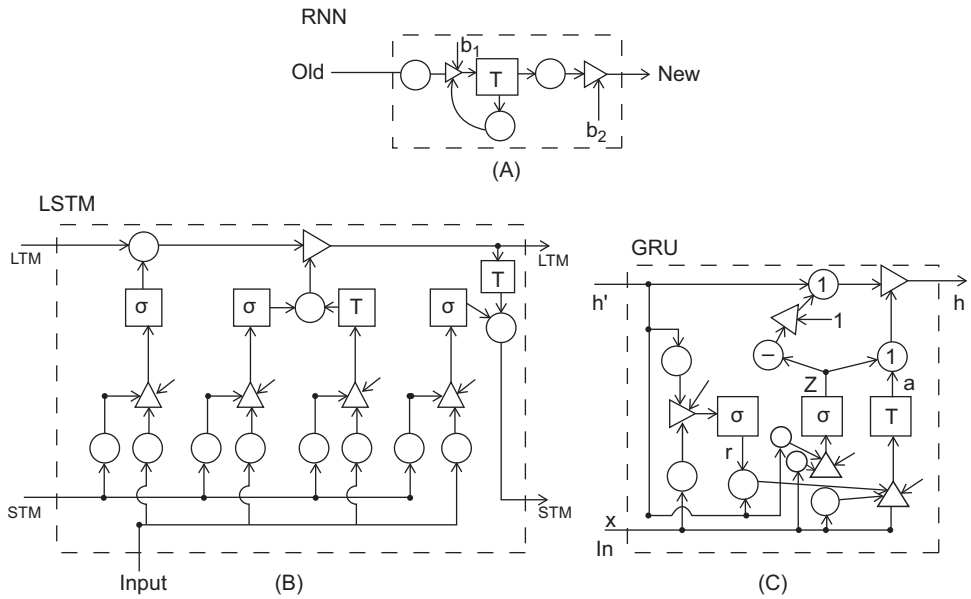


FIGURE 2.8 Types of recurrent neural networks. (A) RNN, (B) LSTM, (C) GRU.

of greater depth than one. In these cases, the feedback connection is voided and new connections are made from a node to another identical node one level of depth below. The output from the weight connects to the new triangle of the underneath node with bias b_1.

Long short-term memory uses two paths instead of the feedback loop utilized in the RNNs. See Fig. 2.8B. One path on top is for long-term memory, and a different path on the bottom is for short-term memory. Both sigmoid and hyperbolic tangent activation functions are employed in LSTM nodes. See Section 2.6 for a description of these activation functions. In the following, a single LSTM node is partitioned into five stages. Each stage contains an activation function. Inputs when modified go through a sigmoid function, all contained in the first stage and often called the forget stage. This stage provides the percent of long-term memory, which should be remembered. The next stage also uses a sigmoid function and is employed to indicate the percent of potential memory to remember. The following stage involves a hyperbolic tangent activation function and addresses potential long-term memory. It must decide how much potential memory to remember. The degree of remembering is controlled by the input. The input gate illustrated by the transmission along the top of the figure refers to how the LSTM unit determines how we should update the long-term memory. The final stage updates short-term memory, by first using the hyperbolic tangent activation function. Here, the potential short-term memory is used, and this stage is called the output gate. The output gate also has inputs, indicating the percent of potential memory to remember from the previous sigmoid activation function. Inputs only go into the first four stages and are utilized in indirect ways by the final stage. The final stage outputs both the long-term memory and the short-term memory. LSTM avoids the exploding and vanishing gradients. However, it is still difficult to train

since in applications there have to be many layers of LSTM nodes. Transfer learning is also not useful in LSTM.

The final illustration is Fig. 2.8C. It is a diagram of a node for a GRU. This diagram provides the architecture corresponding to the equations given by Cho et al. (2014). These equations are provided next, by letting F represent the sigmoid activation function and using T to represent the hyperbolic tangent activation function. In this figure, the input from the bottom is denoted by x, the previous state input is illustrated on the top, and it is h′. Then the leftmost sigmoid output function is denoted by r, where $r = F\ (w_1 \cdot x + w_2 \cdot h' + b_1)$. In a similar way, the next sigmoid has output given by z, where $z = F\ (w_3 \cdot x + w_4 \cdot h' + b_2)$. In this diagram, after z is calculated, it is multiplied by a weight equal to minus one. Then the triangle adds together $1 - z$. This quantity then is multiplied by both weight one and h′. The value z is also multiplied by the output a of the activation function T. Thus, the new state output h is given as a convex combination of the old state h′ and the output of the hyperbolic tangent activation function, a. Here, a slight difference is made between the equations described by the diagram and those in the reference. Specifically, the output state $h = (1 - z) \cdot h' + z \cdot a$ and $a = T\ (w_5 \cdot x + w_6 \cdot (r \cdot h') + b_3)$. In the referenced paper, a different convex combination was used, $h = z \cdot h' + (1 - z) \cdot a$.

References

Berger, M., 1964. Problems in Geometry. Springer-Verlaine.

Carlo, G., Troyer, M., 2022. Solving the quantum many body problem with arterial near networks. Science 355 (6325).

Cho, K., et al., 2014. Empirical evaluation of gated recurrent neural networks on sequence models. ArXiv:1412.3555.

Choi, J., Kim, J., 2020. Tutorial on quantum convolutional neural networks (QCNN). Quantum Phys. .

Graff, C., Giardina, C., 2005. Dynamic Petri-nets: a new modeling technique for sensor networks and distributed concurrent systems. MILCOM .

Hochreiter, S., Schmidhuber, J., 1997. Long short term memory. Neural Comput. 9 (8).

Kavitha, S.S., Kaulgud, N., 2022. Quantum K-means clustering method for detecting heart disease using quantum circuit approach. Soft Comput. 27.

Laarhoven, P., Aarts, E., 1987. Available from: https://doi.org/10.1007/978-94-015-7744-1. Simulated Annealing: Theory and Applications. Reidel, Dordrecht.

LeCunn, Y., LeCun, B., Boser, J., Denker, D., Henderson, R., Howard, W., et al., 1989. Handwritten digit recognition with a back-propagation network, in NIPS'89.

Lang, S., Tate, J., 1958. Principal homogeneous space over abelian varieties. Am. J. Math. 80 (3).

Nielsen, M., Chuang, I., 2000. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

Petri, C., 1962. Kommunikation Mit Automaten (Ph.D. thesis). University of Bonn.

Wittick, P., 2014. Quantum Machine Learning, What Quantum Computing Means to Data Mining. ISBN: 9780128100400.

Wu, M., 2021. A study on arrhythmia via ECG signal classification using the CNN. Front. Comps. Neurosci.

CHAPTER

3

Basic algebras underlying quantum and NN mechanisms

3.1 From a vector space to an algebra

Referring to the arbitrary vector and Hilbert space representations in the MSA will allow generalizations of these structures to be described in a high or global view. To begin, a vector space equipped with a bilinear operation is called an algebra, when several equational identities hold (Gratzer, 1969). The bilinear operation is usually defined by multiplying two vectors resulting in another vector. This multiplication can take numerous forms depending on the carrier set corresponding to the sort of objects being manipulated. The carrier set could result in signature sets involving the point-wise type of multiplication as in function spaces. It could be a composition of operators, that is, functions of functions. It can be the concatenation of equivalence sets consisting of paths as in homotopy. Additionally, it can be convolution or group multiplication as it is often called. The last type of multiplication is the principal operation employed in convolutional neural networks (CNNs). Independent of the type of multiplication, the MSA global view embellishes all of these representations.

More precisely, the sorts for describing an algebra consist of VECTOR and SCALAR, as in the vector space structure. All the corresponding signature sets for vector space and field structures must hold in this case. However, an additional operator name is included. It is a binary operational name and exclusively involves sort VECTOR. The element is named BINE, indicating it is a binary operation and it is such that.

BINE maps VECTOR x VECTOR→VECTOR; as usual, the order of the operands being multiplied together does not matter. Both operands could be on the left or right when multiplied together.

When BINE is included within signature sets among those of a vector space structure, an arity sequence of (1, 1, 3 (2, 1)) results for an algebra. Here, again there are three binary operations, inner product excluded. Two of the three operations belong to the same signature set having names of arity two; they are V-ADD and BINE. The other binary operation name is in a signature set of its own; it is S-MULT. There is a single unary operator name, V-MINUS, and a single zero-ary name, V-ZERO.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00017-5

In an algebra, three additional constraining equations must hold, excluding those for a vector space and a field structure. These constraints are given below. In order to describe these constraints in a more succinct manner, denote:

SCALAR by: a, b
VECTOR by: u, v, w
V-ADD by $+$
S-MULT by $\times$
BINE by $\cdot$

These equational identities for an algebra are as follows:

1) Distributive law: $(u + v) \cdot w = u \cdot w + v \cdot w$.
2) Distributive law: $w \cdot (u + v) = w \cdot u + w \cdot v$.
3) Multiplicative homogeneity: $ab \times (u \cdot v) = (a \times u) \cdot (b \times v)$.

An algebra is said to be associative when the additional constraint (4) holds:

4) Associative law: $((u \cdot v) \cdot w) = (u \cdot (v \cdot w))$.

Additionally, an algebra is said to be unital whenever, along with V-ZERO, there exists V-ONE, both belonging to the zero-ary signature set for VECTOR. Moreover, these elements must be distinct from each other, as indicated by the separate arrows in the polyadic graph (Fig. 3.1). As a consequence, an arity sequence of (2, 1, 3 (2, 1)) exists for a unital algebra. Here, equational identities (1), (2), (3), and (5), given below, must hold true. Again for convenience, replace V-ONE with I,

5) V-ONE law: $I \cdot v = v \cdot I = v$.

An associative algebra with a unital element is a unital associative algebra satisfying all identities mentioned earlier. Moreover, when (6) holds, the algebra is also said to be commutative or abelian.

6) Commutative law: $v \cdot w = w \cdot v$.

When all equational identities hold, a unital algebra is called an associative, commutative unital algebra. A unital associative algebra in which all nonzero elements are invertible is called a division algebra. Again, the polyadic graph of a unital algebra is provided in Fig. 3.1. In this diagram, none of the many-sorted polyadic arrows solely involving

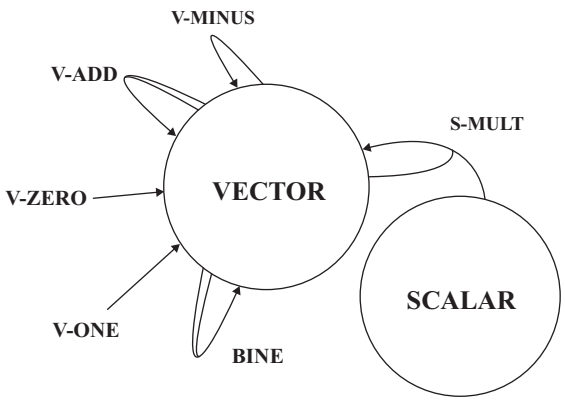


FIGURE 3.1 Polyadic graph of a unital algebra.

SCALAR are displayed. Also, as in a vector space or an R-module, the only link or connection between a VECTOR and SCALAR is through S-MULT. Moreover, the S-MULT operation should use a scalar and a vector in either order. Accordingly, the operation is somewhat commutative, and the result must be the same.

Example 3.1:

Let the carrier set for sort VECTOR be the set of all complex-valued entire functions. Among the many equivalent characterizations of entire functions is that they have a power series representation with an infinite radius of convergence. Using the usual point-wise complex addition and multiplication provides an example of a unital commutative associative algebra. This algebra is important for the development of the holomorphic functional calculus, which is useful in spectral theory mentioned in later sections.

Entire functions are analytic everywhere and so are the sums and products. In general, their reciprocals are entire for nonzero scalars. The unital element, V-ONE, is the number 1. Included in this algebra are all polynomials, sin(z), cos(z), and hyperbolic sine and cosine, along with their sums and products. Not included, for instance, are rational functions having poles in C, $z^{1/2}$ having a branch point at the origin in C, exp(1/z) having an essential singularity in C, and finally any function employing the conjugate z*. However, functions with only removable singularities are in this unital commutative algebra. For instance, sin(z)/z has the singularity at 0, which can be removed.#

Example 3.2:

Consider the carrier set for sort VECTOR being the following: the set M of all 2 by 2 complex-valued matrices, A and B in M. By using the usual matrix addition, multiplication by complex scalars and associated rules results in a vector space structure. The matrix of all zeros = V-ZERO. If the usual matrix multiplication is employed for BINE, then the result of BINE(A, B) = C is another matrix in M. V-ONE is the identity matrix. All the equational identities (1)−(5) for a unital associative algebra hold. It is not a commutative algebra. If S is the subset of all scalar matrices in M, that is, all matrices with zeros off the main diagonal, then it follows that S is a unital commutative associative algebra.#

Example 3.3:

Again consider the carrier set for sort VECTOR being the following: the set M of all 2 by 2 complex-valued matrices. As before, using the usual matrix addition, multiplication by complex scalars and associated rules results in a vector space structure. The matrix of all zeros = V-ZERO. Now instead of the usual multiplication, that is, the usual product C • D of two matrices C and D, let BINE (C, D) = (C • D + D • C)/2. This is similar to the Poisson bracket described in later sections. In fact, this multiplication is central to the Jordan algebra outlined in Section 19.2. The multiplication using BINE might be referred to as a Jordan bracket in this case.

All closure conditions illustrated by the polyadic arrows within the graph of a unital algebra will be shown to hold. The multiplication is closed, since it is a scaled sum of two

usual multiplications. The identity V-ONE is I. However, to prove that this is an algebra, it must be shown that this type of multiplication satisfies the identities (1), (2), and (3) in the definition of an algebra. These restrictions are proven next:

1) Distributive law:
Note that BINE ((E + F), G) = ((E + F) • G + G • (E + F))/2 = ((E • G + F • G + G • E + G • F))/2 = (E • G + G • E)/2 + (F • G + G • F)/2 = BINE (E, G) + BINE (F, G).
2) Distributive law: Similarly holds
3) Multiplicative Homogeneity:

BINE (a × E, b × F) = (a × E • b × F + b × F • a × E)/2 = ab × BINE (E, F) holds.
So this structure is an algebra.

6) Commutative law:
It is also commutative since BINE (E, F) = BINE (F, E) holds.
Moreover, it is unital because if I is the 2 by 2 identity matrix, then it too is the identity for the algebra, that is, it satisfies the equational constraint (5), since:
5) V-ONE law:
I • v = v • I = v, and BINE (I, E) = (I • E + E. • I)/2 = BINE (E, I) = (E • I + I • E)/2 = E.
However, the associative law does not hold as seen below. With this sort of multiplication and ignoring the vector space, the multiplicative structure is only a groupoid and not a semigroup,
4) Associative Law:

First consider BINE (BINE (E, F), G) = BINE ((E • F + F • E)/2, G) = (E • F • G + F • E • G + G • E • F + G • F • E)/4. Second consider BINE (E, BINE (F, G)) = BINE (E, (F • G + G • F)/2) = E • F • G + E • G • F + F • G • E + G • F • E)/4. The realist is that the first and second equations do not agree. Accordingly, the overall structure is a unital, commutative, nonassociative algebra.#

As previously noted, the last example will be mentioned again in reference to Jordan algebras and reproducing kernel Hilbert spaces in Section 19.2.

3.2 An algebra of time-limited signals

The inner product space of time-limited signals A, as described in Section 1.9, will be shown to form an algebra. For f and g in A, the point-wise convolution is defined as $\mathrm{BINE}(f,g)(n) = (f \star g)(n) = \sum_{k=-\infty}^{\infty} f(n-k)g(k) = \sum_{k=-\infty}^{\infty} g(n-k)f(k)$. With this type of multiplication BINE, the vector space A becomes a unital, commutative, associative algebra. The unital function is $I = (1)_0$, and $I \star f = f \star I = f$. Although the limits in the summation mentioned earlier are infinite, there are only a finite number of nonzero terms. Next, a more concise expression for the actual summation limits will be described.

Recall, for any f in A, the cozero set is defined to be the set of all points in Z for which f is nonzero. This set is most important because COZ(f ⋆ g) is a subset of the dilation of the two sets COZ(f) and COZ(g) (Giardina, 1985). The dilation is given by D(COZ(f), COZ

(g)) = the union of all sets of integers {n + k}, where n is in COZ(f) and k is in COZ(g). The convolution of f and g will be zero outside this dilated set. Therefore, the convolution can be calculated using the following:

$$(f \star g)(n) = \sum_{k \in COZ(f), (n-k) k \in COZ(g)} f(k)g(n-k)$$

An example will illustrate the use of this point-wise formula.

Example 3.4:

Consider the two signals f and g in A given by bound vectors: $f = (3\ 2\ 1)_0$, and $g = (2{-}1)_0$. Then, COZ(f) = {0, 1, 2} and COZ(g) = {0, 1}. Taking the union of all possible sums from these sets and removing duplicates renders the dilated set: {0, 1, 2, 3}. The convolution of f and g will be zero outside of this dilated set. Indeed, $f \star g = (6\ 1\ 0{-}1)_0$, and it is zero outside of {0, 1, 2, 3}. A quick check to see that $(f \star g)(2) = 0$ is given next. To verify this, first notice that n = 2 is in the dilated set. Next, find k in COZ(f) such that n-k = 2 - k is in COZ(g). This occurs for k = 2 and k = 1. Substitute these numbers into the point-wise convolution formula: $(f \star g)(2) = f(2)\ g(0) + f(1)\ g(1) = 1 \cdot 2 + 2 \cdot (-1) = 0$.#

The convolution is more easily determined using the parallel algorithm illustrated in Fig. 3.2 (Giardina, 1988). This algorithm utilizes two operations mentioned in the MSA described in Section 1.7 and illustrated in the polyadic graph in Fig. 1.2. These operations are V-ADD and S-MULT. However, in this application, the operators have bound vectors as arguments. They are not point-wise operations. An additional operation specified in this diagram is TRAN, which is similar to an affine-type operation. The operation TRAN(f; k) translates the (whole) bound vector k units. It translates to the right for k > 0, to the left for k < 0, and not at all for k = 0. Thus, for $f = (a_0, a_1, \ldots a_m)_n$, it follows that TRAN(f; k) = $(a_0, a_1, \ldots a_m)_{n+k}$. Accordingly, with this operation, values within the bound vector stay the same, but the vector is shifted to k units. Looking at it differently, the point n in the subscript becomes the point n + k by an additive group action on f by using elements of the cozero set COZ(g).

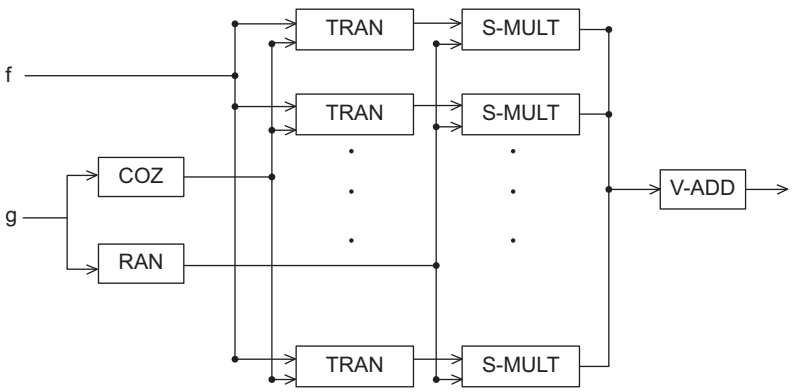


FIGURE 3.2 Parallel convolution algorithm.

Momentarily, in Fig. 3.2, it is convenient to view the cozero set for g as a stack, denoted by coz(g). Simultaneously, view the data structure, ran(g) as a stack with values of g corresponding in order to those integers in coz(g). So ran(g) is like the nonzero range of g. Because of this arrangement, when the stacks are popped in unison, the function g is revealed. Consider the Example 3.5 as an illustration.

Example 3.5:

Let the two signals f and g in A be given by bound vectors: $f = (3\ 2\ 1)_0$, and $g = (2{-}1)_0$. Th objective is to find the convolution h. In this case, coz(g) and ran(g) are, respectively, illustrated as stacks:

```
| 0 |        | 2 |
|_1_|        |_−1_|
```

When these stacks are popped, values in ran(g) provide scalar values used in S-MULT. Simultaneously, popping coz(g) specifies the number of units f will be translated; this is the group action mentioned previously. An application of the algorithm first gives a scalar multiplication by two along with a zero translation. Next, this is followed by a scalar multiplication by minus one and a translation of one. Finishing with a vector addition, V-ADD, gives $h = \text{V-ADD}(2f, -(3\ 2\ 1)_1) = (6\ 4\ 2\ 0)_0 + (0{-}3\ -2{-}1)_0 = (6\ 1\ 0{-}1)_0$.#

Similar to standard convolution, dilated convolution can be employed in CNNs. Convolution can be created point-wise as well as in parallel. Again, the algorithm depicted in Fig. 3.2 can be applied with minor modifications to perform dilated convolution. In this type of convolution, one signal g called the filter is dilated, and the other, the data signal f, is utilized as is. All this means that zeros are padded or placed in between the entries within g, resulting in another bound vector, g′. The number of zeros is usually a small positive integer r. The point-wise convolution in this case is $(f \circledast g')(n) = \sum f(k)g'(n-k)$. As usual, the sum should be over k in COZ(f) and over n-k in COZ(g′). Specifically, for the original bound vector, $g = (g(k)\ g(k+1)\ \ldots\ g(k+p))_j$; the bound vector $g' = (g(k)\ 0\ ..\ 0\ g(k+1)\ 0\ \ldots\ 0\ g(k+p))_j$ is used in the convolution formula. An example should make this clearer.

Example 3.6:

Let $f = (2\ 1\ 0{-}1)_0$ and $g = (2{-}1)_0$; then use r = 1; that is, a single zero will be inserted between the values of g starting with the subscript integer. In this case, $g' = (2\ 0{-}1)_0$, and the convolution result is $(f \circledast g') = (4\ 2{-}2\ -3\ 0\ 1)_0$. To validate at a single point, say n = 3 use the pointwise convolution, at the point n = 3, ⊛ and obtain $(f \circledast g')(3) = -3$. To see this, use k in COZ(f) and 3-k in COZ(g′). This holds for k = 1 and k = 3. So $(f \circledast g')(3) = f(1)g'(2) + f(3)g'(0) = -1{-}2 = -3$. Additionally, the parallel algorithm can be applied just as before. This time, using sets COZ(g′) and RAN(g′) in place of stacks, these ordered sets are (0, 2) and (2, −1), respectively. Using elements from these sets in the order specified gives $f \circledast g' = \text{V-ADD}\ [2f, -1(2\ 1\ 0{-}1)_2] = (4\ 2\ 0\ {-}2\ 0\ 0)_0 + (0\ 0\ {-}2\ {-}1\ 0\ 1)_0 = (4\ 2{-}2\ {-}3\ 0\ 1)_0$.#

In stride-type convolution often employed in CNNs, both the data signal f and the filtering signal g are dilated. This means that a fixed number usually r = 1, 2, or 3 zeros is placed in between entries for f and for g. The same value r is used for both, resulting in new bound vectors f′ and g′. Then the convolution of f′ and g′ proceeds as before.

3.3 The commutant in an algebra

Algebras have been studied a long time in quantum mechanics, particularly on the commutation rules. (McCoy, 1929). For a subset B of a noncommutative algebra A, the concept of a commutant Bc is useful. It is defined as the subset of A that commutes with every element of B. The subset Bc = {a in A | a • b = b • a for all b in B}. The concept of commutant is illustrated in the following example.

Example 3.7:

Again, consider the carrier set for sort VECTOR being the following: the set A of all 2 by 2 complex-valued matrices. Now employ the usual matrix addition scalar multiplication as well as the usual matrix multiplication. It was seen that this structure forms a unital associative algebra, which is not commutative. To illustrate the commutant operation, use the subset B of A consisting of 2 by 2 matrices with all zero elements except for the first row and second column location. That is, matrix entry b12 is arbitrary. The commutant Bc in this case is the set of all 2 by 2 matrices such that a11 = a22 and a21 = 0. Also, a12 is an arbitrary complex number. A typical matrix in Bc is

$$\begin{vmatrix} c & a \\ 0 & c \end{vmatrix}.$$

Of more interest in this example is the fact that Bc itself is also an associative algebra, with identity, a11 = a22 = 1, a12 = a21 = 0; therefore, it is unital. Moreover, this time, it is commutative. So, if C and D are matrices in Bc, then C • D = D • C. The commutant will be seen again in the von Neumann algebra in a later chapter.#

The commutant is currently heavily studied in relation to Hilbert space fragmentation and ergodicity (Moudgalya, 2022). The final fact involving algebras, for now, is that sometimes algebras are defined as bilinear maps over a module, not over a vector space. An instance of this is the Grassmann algebra also to be described later.

3.4 Algebra homomorphism

An algebra homomorphism involves two algebras S and M, possibly unital, associative, or commutative over a field, say C. The homomorphism is the map ϕ, where ϕ: A → B and for any A and D in S, B in M, and c in C, the following hold:

1) Additive: ϕ (A + D) = ϕ (A) + ϕ (D).
2) Multiplicative: ϕ (A D) = ϕ (A) ϕ (D).

3) Scalar: ϕ (c A) = c ϕ (A).
4) Identity: ϕ (I_A) = I_B, when there exist unital elements.

Example 3.8:

Consider again the carrier set for VECTOR of all 2x2 complex-valued matrices and for SCALAR let the carrier set be C. Example 3.2 illustrates that this structure M is a unital associative algebra using the usual matrix operations. If S is the subalgebra M, then for every invertible matrix, B in M, and A, and D in S, $\phi(A) = B^{-1}A\ B$ is a unital algebra homomorphism. This follows since:

1) Additive: ϕ (A + D) = B^{-1}(A + D) B = B^{-1}A B + B^{-1}D B) = ϕ (A) + ϕ (D).
2) Multiplicative: ϕ (A D) = B^{-1}(A D) B = B^{-1}A B B^{-1}D B = ϕ (A) ϕ (D).
3) Scalar: ϕ (c A) = B^{-1}c A B = c B^{-1}A B = c ϕ (A).
4) Identity: ϕ (I) = B^{-1}I B = B^{-1}B = I.#

When a homomorphism is one-to-one and onto, it is called an algebra isomorphism.

3.5 Hilbert space of wraparound digital signals

Different types of digital signals give rise to carrier sets for which deeper structures apply than for those signals in A. Recall that A consists of all digital signals of compact support in R^z. Now, wraparound digital signals will be described with wraparound convolution. This structure forms a Hilbert space, as well as an associative, commutative, unital algebra. In this case, sort SCALAR again represents the real number field. While VECTOR symbolizes bound vectors f in R^{Zn}, here f: $Zn \rightarrow R$, where Zn is the cyclic group consisting of {0, 1, 2, ..., n-1}, n > 1. This set of integers can be thought to lie on a circle, so the successor of n − 1 is 0. The successor of 0 is 1, the successor of 1 is 2, etc. The earliest instance of a cyclic group in this text is in Example 1.1. In this case, the additive group in the Galois field is mod three.

A bound vector in this environment will always be written as f = (a b c ... d)$_k{}^{Wn}$, where a, b, ... d are n real numbers. Accordingly, V-ZERO = (0 0 ... 0)$_k{}^{Wn}$. For wraparound bound vectors, the zero vector is always specified in this manner. Additionally, *Wn* signifies that this vector is a wraparound bound vector in R^{Zn} and k is an integer between 0 and n − 1. Using point-wise addition, as well as convolution for the multiplication of these types of bound vectors, results in a unital, commutative, and associative algebra structure. The unital bound vector V-ONE is (1 0 ... 0)$_0{}^{Wn}$ in R^{Zn}. Convolution for bound vectors in R^{Zn} is given by essentially the same algorithms as previously described for bound vectors in R^Z: As usual, convolution with the zero vector yields a zero vector. Thus, in the following, assume that the bound vectors are nonzero.

The point-wise convolution is given by:

$$(f \star g)(n) = \sum_{k \in COZ(f),(n-k) \in COZ(g)} f(k)g(n-k)$$

Wraparound convolution is a form of full convolution, wherein the resulting convolution has the same size as the input signal. Previously described time-limited digital signal convolution might be called extended convolution. The same parallel algorithm illustrated in Fig. 3.2 can be used in R^{Zn}. Here, the names of the operations are the same, but since the carrier sets differ the actual operations are distinct. An example to make this clearer is given next. The cyclic group acts on the vector space structure using TRAN, much like vectors act on affine elements in affine spaces. Again, since the MSA global view is the same as in R^{Z}, the actual lower level operations are different.

Example 3.9:

Let $f = (2{-}1\ 3\ 0\ 1)_0{}^{W5}$ and $g = (1{-}1\ 2\ 0\ 1)_0{}^{W5}$. Similarly, COZ(g) = {0, 1, 2, 4} and RAN (g) = {1−1, 2, 1}. The parallel convolution algorithm yields the following: $f \star g = 1 \cdot f - 1 \cdot (2{-}1\ 3\ 0\ 1)_1{}^{W5} + 2 \cdot (2{-}1\ 3\ 0\ 1)_2{}^{W5}\ 1 \cdot (2{-}1\ 3\ 0\ 1)_4{}^{W5} = f - (1\ 2{-}1\ 3\ 0)_0{}^{W5} + 2\ (0\ 1\ 2{-}1\ 3)_0{}^{W5} + (-1\ 3\ 0\ 1\ 2)_0{}^{W5} = (0\ 2\ 8{-}4\ 9)_0{}^{W5}$. Now, use the point-wise formula in order to verify that $(f \star g)(3) = -4$. As before, employ values of k in COZ(f) while 3-k is in COZ(g). These values are given by k = 1, k = 2, and finally k = 4. The last value applies because 3−4 = −1 = 4 modulo(5). That is, the tuple number four in R^{Zn} is the same as the minus first tuple, if such a tuple existed. Accordingly, $(f \star g)(3) = f(1)g(2) + f(2)g(1) + f(4)g(4) = (-1) \times 2 + 3 \times (-1) + 1 \times 1 = -4$.#

The vector space R^{Zn} also forms an inner product space, again using a dot product on the nonempty intersection of cozero sets. And it is zero otherwise. Section 1.9 provides the axioms, and Fig. 1.3 illustrates a polyadic graph for the MSA representation of this structure. The inner product of $f = (a\ b\ c\ \ldots\ d)_k{}^{Wn}$ and $g = (e\ h\ \ldots\ m)_k{}^{Wn}$, as usual, with all n entries shown is $= a \cdot e + b \cdot h + \ldots + d \cdot m$. The inner product is again represented by $<f, g>$ or $<f|g>$. If the complex field were employed instead of the real field, the values a, b, …, d need to be conjugated when the inner product is evaluated. This is the convention used in quantum physics and illustrated in Section 1.8. The inner product in R^{Zn} induces the two norm mentioned earlier. This norm can be found from the formula $\|f\|^2 = <f|f>$. Under this norm, R^{Zn} is a complete inner-product space and therefore a Hilbert space; see Appendix A.1 for the proof. Convergence of sequences of bound vectors in R^{Zn} is similar to convergence in R^n.

3.6 Many-sorted algebra description of a Banach space

A Banach space is a vector space with a norm, over the complex or real field. A few norms have already been illustrated. The two norm was induced by the inner product, see Section 1.8. Also, the one norm was employed in the k-means unsupervised learning technique in Example 2.1 in Section 2.1. A Banach space is similar to a Hilbert space in that it too is complete with respect to the norm. As usual, the norm is a translational invariant-type distance function. However, unlike a Hilbert space, separability is not sufficient to guarantee the existence of a Schauder basis (Enflo, 1973). Accordingly, a Schauder basis will be assumed when working in an infinite dimensional Banach space. Otherwise, an

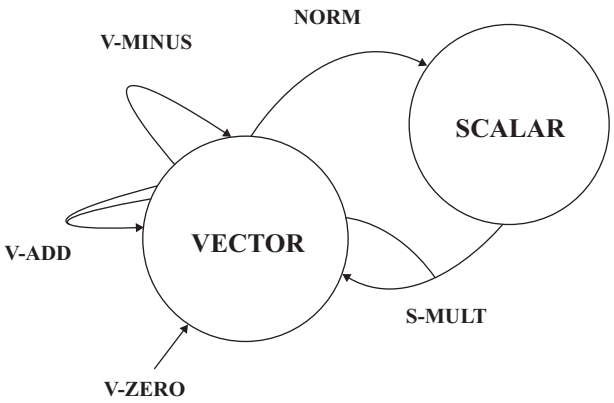


FIGURE 3.3 Polyadic graph for a Banach space.

unconditional basis needs to be employed as in general topological vector spaces. A MSA global view will be given next for a Banach space.

In the MSA description, the Banach space entity consists of two sorts, and they are SCALAR and VECTOR. All the signature sets associated with SCALAR are the same as those for a real or complex field structure. The signature sets for the VECTOR sort are exactly the same as those for a vector space, except that there exists an additional unary operation name: NORM, within its own signature set. From this, it follows that the arity sequence for a Banach space is (1, 2(1 1), 2(1 1)). Fig. 3.3 illustrates the NORM operational name used in a Banach space, along with the usual vector space operational names within signature sets. Operational names exclusively associated with SCALAR are omitted. The NORM is such that:

$$\text{NORM: VECTOR} \rightarrow \text{SCALAR}$$

The vector space operators illustrated are as follows:

$$\text{V} - \text{ADD:VECTOR x VECTOR} \rightarrow \text{VECTOR}$$

$$\text{S} - \text{MULT: SCALAR x VECTOR} \rightarrow \text{VECTOR}$$

$$\text{V} - \text{MINUS: VECTOR} \rightarrow \text{VECTOR}$$

$$\text{V} - \text{ZERO, 0 Element in VECTOR}$$

As usual, the S-MULT operation must allow scalars on both sides of the vector in the multiplication operation.

The NORM must satisfy three conditions. To see this, first replace:

SCALAR by a
VECTOR by u, v, w
S-MULT by $\cdot$
V-ADD by $+$
NORM by $\|\|$

The conditions for NORM to satisfy are the following:

1) Positive definite: $\|u\| > 0$, and $\|u\| = 0$, iff $u = 0$.
2) Homogeneous: $\|a \cdot u\| = |a| \|u\| = \|u \cdot a\|$.
3) Triangle inequality: $\|u + v\|$ is less than or equal to $\| u \| + \| v \|$.

The norm in a Banach space B is continuous. This follows since for T and T_n in B, if $T_n \rightarrow T$, then by the triangle inequality, $| \|T\| - \|T_n\| |$ is less than or equal to $\|T - T_n\|$.

There are numerous important examples of Banach spaces. They include sequence spaces defined on the natural numbers, as well as spaces of continuous functions and operators.

Example 3.10:

Consider the space of continuous functions on the closed interval zero to one: B = C([0, 1]) with norm of g, $\|g\| = \max |g|$ for x in [0, 1]. This is the set of all continuous functions on a compact, which is a closed and bounded set on the real line. All the axioms associated with a Banach space hold. For instance, the sum of two continuous functions is continuous, so is the scalar product. The given norm satisfies all the conditions of a normed vector space. So, as an illustration using f also in B, it follows that:

3) Triangle inequality: $\|f + g\| = \max |f + g|$ is less than or equal to $\max (|f| + |g|)$, which is less than or equal to $\max |f| + \max |g| = \|f\| + \|g\|$.#

3.7 Banach algebra as a many-sorted algebra

A Banach algebra is a unital associative algebra, as well as Banach space over the real or complex field (Larsen, 1973). The arity sequence for this structure is (2, 2(1, 1), 3(2, 1)). Since it is a unital associative algebra, it has a V-ONE and a vector-type multiplication

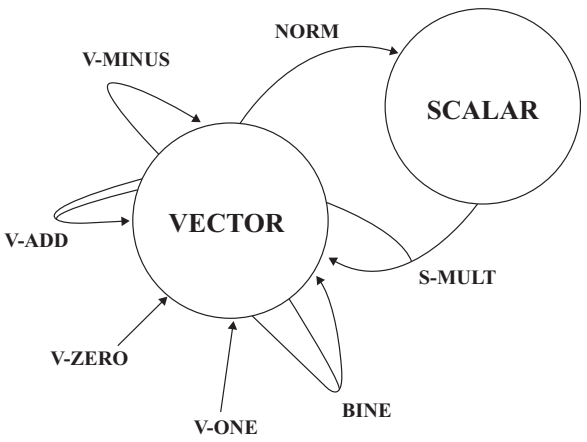


FIGURE 3.4 Polyadic graph for operator names in a Banach algebra.

named BINE, (Fig. 3.4). However, this operation need not be commutative. When it is commutative, the algebra is called a commutative Banach algebra.

Since a Banach algebra is also a Banach space, it satisfies all three equational identities for a Banach space above. In addition, it has to satisfy a fourth condition mentioned below. This condition links the NORM operational name with the binary operational name BINE. The latter is a vector multiplication name symbolized by $(\cdot)$. The constraint is given in equation (4) below. The triangle product inequality-type constraint makes the multiplication a continuous operation.

4) Triangle Product Inequality: $\|v \cdot w\|$ is less than or equal to $\|v\|\ \|w\|$.

As mentioned earlier, BINE is a continuous operation in a Banach algebra, B. To show this, if $T_n \to T$, $S_n \to S$, and also T_n, T, S_n, and S in B, then, $T_n\ S_n \to T\ S$. This follows since $\|T_n\ S_n - T\ S\|$ is less than or equal to $\|(T_n - T)\ S\| + \|T(S_n - S)\ \|$. Because S and T are bounded, then $\|(T_n - T)\ S\| + \|T(S_n - S)\ \|$ is less than or equal to $\|(T_n - T)\|\ \|S\| + \|T\|\ \|(S_n - S)\| \to 0$.

A polyadic graph is given in Fig. 3.4, illustrating the names of operators in a Banach algebra. Not shown in this diagram are the names of operators exclusively involving sort SCALAR.

Example 3.11:

The carrier sets for VECTOR, as well as SCALAR, use the field of all complex numbers. This vector space becomes a commutative Banach algebra over the complex field with the NORM $(z) = |z|$. As usual, for $z = x + iy$, $|z|^2 = x^2 + y^2$. #

3.8 Many-sorted algebra for Banach* and C* algebra

A Banach* algebra is a Banach algebra over the real or complex field with an additional unary operation. The resulting arity sequence is (2, 3(2, 1), 3(2, 1)). The additional operation is an involution similar to conjugation. Two applications of this unary function result in the identity; it is an involution. Additionally, five equational identities must hold for the Banach* algebra structure; these are described below.

For a description of the Banach* algebra structure in a MSA high view, begin with the existence of two sorts; they are SCALAR and VECTOR. The SCALAR sort has a corresponding carrier set consisting of the complex field, and all operations are those of a complex field. The real field is also applicable. The signature sets for VECTOR sort are exactly the same as in a Banach algebra, except that there is an additional unary operational name: ADJ. Here,

ADJ: VECTOR → VECTOR.

In order to provide the equational identities that must hold, replace the sorts by symbols and denote operator names by suggestive characters:

VECTOR by v, w
SCALAR by a
S-MULT by $\times$

V-ADD by +
V-ONE by I
BINE by ·
NORM by $\| \ \|$
ADJ by *

The five laws that must hold are the following:

1) Involuntary: $(v^*)^* = v$.
2) Additive: $(v + w)^* = v^* + w^*$.
3) Conjugate linear: $(a \times v)^* = \bar{a} \times v^*$, Here, $\bar{a}$ denotes the conjugate of a.
4) Transpose: $(v \cdot w)^* = w^* \cdot v^*$.
5) Isometry: $\|v^*\| = \|v\|$.

The most important type of Banach* algebra is the C* algebra. It is a Banach* algebra also satisfying:
6) C* Identity: $\|v \cdot v^*\| = \|v\|^2$.

Fig. 3.5 provides an illustration of the operations in a C* algebra using a polyadic graph. Not shown in this diagram are operations solely involving SCALAR.

3.9 Banach* algebra of wraparound digital signals

The inner product in R^{Zn} induces the two norm mentioned earlier. Under this norm, R^{Zn} is a complete inner-product space and therefore a Hilbert space. Since R^{Zn} is a Hilbert space, it is also a Banach space using the two norm.

Also, using the one norm for f in R^{Zn}, it is also a Banach space. Moreover, it is a Banach algebra as described using the MSA in Section 3.7. The illustration given in the Fig. 3.4 provides a polyadic graph for this structure. The Banach algebra structure is a result of the

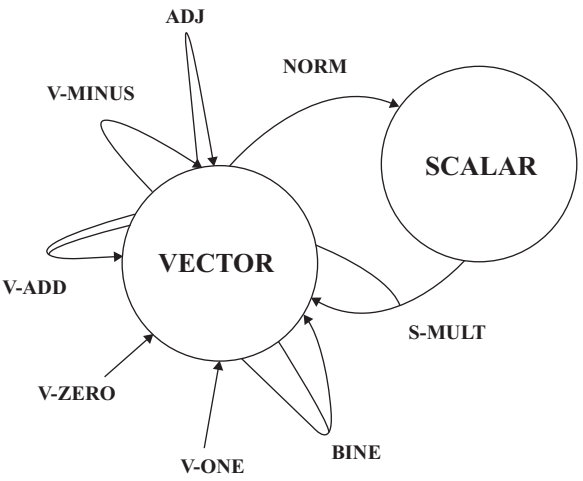


FIGURE 3.5 Polyadic graph of operators in a Banach* or C* algebra.

equational identity #4, the triangle product inequality holding for convolution in R^{Zn}. In this case, $\|f \star g\|_1 = \sum_n |(f \star g)(n)| = \sum_n |\sum_k f(k)g(n\text{-}k)|$ is less than or equal to the sum: $\sum_n \sum_k |f(k)|\,|g(n\text{-}k)| = \sum_k \sum_n |f(k)\|g(n\text{-}k)| = \sum_k |f(k)| \sum_n |g(n\text{-}k)| = \sum_k |f(k)|\ \|g\|_1 = \|f\|_1\|g\|_1$. In general, the inequality holds. As an illustration, using the one norm on the results from a previous example, see Examples 3.9 and 3.12:

Example 3.12:

Let $f = (2{-}1\ 3\ 0\ 1)_0{}^{W5}$ and $g = (1{-}1\ 2\ 0\ 1)_0{}^{W5}$, $f \star g = (0\ 2\ 8{-}4\ 9)_0{}^{W5}$. Now illustrate that $\|f \star g\|_1$ is less than or equal to $\|f\|_1\|g\|_1$. This is the triangle product inequality dealing with a Banach algebra. Adding the absolute values of entries within the convolution above gives $\|f \star g\|_1 = 23$; while $\|f\|_1 = 7$ and $\|g\|_1 = 5$, the product $\|f\|_1\|g\|_1$ is 35.#

From the proof mentioned earlier, note that equality will hold in the triangle product inequality whenever all values within bound vectors f and g are nonnegative.

Example 3.13:

Let $f = (2\ 1\ 3\ 0\ 1)_0{}^{W5}$ and $g = (1\ 1\ 2\ 0\ 1)_0{}^{W5}$, $f \star g = \text{V-ADD}(f, g) = f + (1\ 2\ 1\ 3\ 0)_0{}^{W5} + 2\ (0\ 1\ 2\ 1\ 3)_0{}^{W5} + (1\ 3\ 0\ 1\ 2)_0{}^{W5} = (4\ 8\ 8\ 6\ 9)_0{}^{W5}$. Here, $\|f \star g\|_1 = 35$. As before, $\|f\|_1\|g\|_1 = 35$.#

Finally, applications will be made for the Banach* algebra and a C* algebra, as defined in Section 3.8., as well as the polyadic graph for a Banach* algebra and a C* algebra, which are provided in Fig. 3.5. The space R^{Zn} is a Banach* algebra. This trivially holds true since the unary operator name in Fig. 3.5 is ADJ. The adjoint or conjugate of a real number equals itself. The next example, however, shows that R^{Zn} is not a C* algebra; that is, equation (6), for the C* Identity: $\|v \cdot v^*\| = \|v\|^2$, does not hold. In this case, $\|f \star f^*\|_1 = \|f \star f\|_1$ does not equal $\|f\|_1{}^2$.

Example 3.14:

Consider $f = (1{-}1\ 1)_0{}^{W3}$ in R^{Z3}. Then $f \star f = f - (1\ {-}1\ 1)_1{}^{W3} + (1\ {-}1\ 1)_2{}^{W3} = f - (1\ 1\ {-}1)_0{}^{W3} + (-1\ 1\ 1)_0{}^{W3} = (-1\ {-}1\ 3)_0{}^{W3}$. Therefore $\|f \star f\|_1 = 5$. On the other hand, $\|f\|_1{}^2 = 9$. So this Banach* algebra is not a C* algebra.#

It is interesting to see that there are situations where $\|f \star f\|_1 = \|f\|_1{}^2$ even when f does not possess all nonnegative values.

Example 3.15:

If $f = (-2\ 0\ 3\ 0{-}1)_0{}^{W5}$, $f \star f = -2\ \text{TRAN}(f; 0) + 3\ \text{TRAN}(f; 2) - 1\ \text{TRAN}(f; 4) = (4\ 0{-}6\ 0\ 2)_0{}^{W5} + (0{-}3\ {-}6\ 0\ 9)_0{}^{W5} + (0{-}3\ 0\ 1\ 2)_0{}^{W5} = (4{-}6\ {-}12\ 1\ 13)_0{}^{W5}$. So $\|f \star f\|_1 = 36 = \|f\|_1{}^2$.#

3.10 Complex-valued wraparound digital signals

It is instructive to illustrate the use of the complex field structure before quantum applications are provided. In this direction, consider C^{Zn}; these are bound signals with complex

entries, and they too form a Banach* algebra, as well as a Hilbert space; for the proof, see Appendix A.1. The conjugation operation for Banach* and C* algebras is more important in this instance. The following example illustrates the conjugation operation in the inner product operations. The convolution of functions is also found along with the convolution involving conjugate functions. The results can be compared to illustrate the five equational identities needed to describe a Banach* algebra.

Example 3.16:

Consider the two bound vectors f and g in C^{Zn} with $i = \sqrt{-1}$, $f = (2\ 0\ 3i\ 2 + i\ 0)_0{}^{W5}$, and $g = (2i\ 0\ 3 + 2i\ 0\ i)_0{}^{W5}$. In order to find f★g, the same parallel algorithm can again be used. Indeed, $f \star g = g \star f = 2TRAN(g; 0) + 3i\ TRAN(g; 2) + (2 + i)\ TRAN(g; 3) = (4i\ 0\ 6 + 4i\ 0\ 2i)_0{}^{W5} + (0{-}3\ {-}6\ 0{-}6 + 9i)_0{}^{W5} + (4 + 7i\ 0\ {-}1 + 2i\ {-}2 + 4i\ 0)_0{}^{W5} = (4 + 11i\ {-}3\ {-}1 + 6i\ {-}2 + 4i\ {-}6 + 11i)_0{}^{W5}$. Now the conjugate of the convolution $f \star g$ is $(f \star g)^* = (4{-}11i\ {-}3{-}1{-}6i\ {-}2{-}4i\ \text{-}6{-}11i)_0{}^{W5}$.

Next the convolution of f* and g* will be determined. First, $f^* = (2\ 0{-}3i\ 2 - i\ 0)_0{}^{W5}$ and $g^* = (-2i\ 0\ 3{-}2i\ 0\ -i)_0{}^{W5}$; then, $f^* \star g^* = -2j\ TRAN(f^*;0) + (3{-}2i)\ TRAN(f^*;2)$ - i TRAN $(f^*;4) = (\ {-}4i\ 0{-}6\ {-}2{-}4i\ 0)_0{}^{W5} + (4{-}7i\ 0\ 6{-}4i\ 0{-}6{-}9i)_0{}^{W5} + (0{-}3\ {-}1{-}2i\ 0{-}2i)_0{}^{W5} = (4{-}11i\ {-}3{-}1{-}6i\ {-}2{-}4i\ \text{-}6{-}11i)_0{}^{W5}$. So, $(f \star g)^* = f^* \star g^*$, which illustrates property #4 from a Banach* algebra.

Next the inner product for a complex Hilbert space will be found, along with other norm calculations. First, it follows that $< f, g > = < (2\ 0\ 3i\ 2 + i\ 0)_0{}^{W5}, (2i\ 0\ 3 + 2i\ 0\ i)_0{}^{W5} > = 4i + (-3i)\ (3 + 2i) = 4i - 9i + 6 = 6{-}5i$. Finally, $f \star f^* = 2\ f - 3i\ (2 + i\ 0\ 2\ 0\ 3i)_0{}^{W5} + (2 - i)\ (3i\ 2 + i\ 0\ 2\ 0)_0{}^{W5} = (4\ 0\ 6i\ 4 + 2i\ 0)_0{}^{W5} + (3{-}6i\ 0{-}6i\ 0\ 9)_0{}^{W5} + (3 + 6i\ 5\ 0\ 4{-}2i\ 0)_0{}^{W5} = (10\ 5\ 0\ 8\ 9)_0{}^{W5}$. So, $\|f \star f^*\|_1 = 32$ and $\|f\|_1{}^2 = (5 + 5^{1/2})^2$. This illustrates that the triangle product inequality property number (4) for a Banach algebra holds. It also shows that the C* algebra property number (6) does not hold.#

In C^{Zn}, the triangle product inequality for a Banach algebra holds in general, as before. Additionally, the transpose property holds in general for a Banach* algebra. This is property number (4), Section 3.7, accordingly: $(f \star g)^* = [\sum_n (f \star g)(n)]^* = [\sum_n (g \star f)(n)]^* = \sum_n (g^* \star f^*)(n) = g^* \star f^*$. As a consequence, C^{Zn} is a Banach* algebra, but not a C* algebra.

References

Enflo, P., 1973. A counterexample to the approximation problem in Banach spaces. Acta Math. 130 (1).

Giardina, C., 1988. Parallel Algorithms for Multidimensional Signal Processing, 22nd Asilomar Conference Proceedings.

Giardina, C.R., 1985. Amany sorted algebra based signal processing data flow language. Int. Conf. Math. Model., 5th .

Gratzer, G., 1969. Universal Algebra. Van Nostrand Reinhold, N.Y.

McCoy, N., 1929. On commutation rules in the algebra of quantum mechanics, U of Iowa. PNAS 15 (3), 200–202.

Moudgalya, S., 2022. Hilbert Space Fragmentation and Commutant Algebras. California Institute of Technology, Benasque.

This page intentionally left blank

CHAPTER

4

Quantum Hilbert spaces and their creation

4.1 Explicit Hilbert spaces underlying quantum technology

In the previous chapter, digital signals and wraparound digital signals were presented as inner product space and Hilbert space, respectively. The basic axioms of quantum are vague in defining the type of underlying Hilbert space. Again at a lower level, a precise specification relative to the global Hilbert space structure shall be made. Here, the actual elements are provided; specifically, carrier sets for the two sorts, one corresponding to SCALAR and the other for VECTOR, are given. The actual operations themselves will be specified matching the names of operations within the signature sets.

Example 4.1:

This is perhaps the simplest example of a Hilbert space. Here, the reals form a vector space, V, over the real number field, R. This fact was previously mentioned in Example 2.13, illustrating the effect of a vector on an element in affine space. The carrier set for SCALAR and the carrier set for VECTOR both consist of elements from the real line. The real numbers play a dual role with the usual operations and functions involving real numbers. All the axioms are upheld for an inner product space, as well as for a Hilbert space. The space is only a single dimension. Note that IN-PROD results in just the product of any two vectors, which in this case are two scalars. The Hilbert space equational constraint (9) holds. See Section 1.8:

9) Positive definite: v • v is greater or equal to 0, and = 0 iff v = 0.#

Another concrete example is provided next.

Example 4.2:

This example is very similar to the previous example. Indeed, the complex number system forms a vector space over the complex field C. It follows that both sorts are complex in this case. Again both carrier sets are the same; this time, they are the complex numbers.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00001-1

Also, see Example 1.7. This structure also forms a very important Hilbert space for Fock space. It describes the vacuum space. Fock space will be presented in the many-sorted algebra (MSA) description in a later chapter. Again, notice that IN-PROD of v with w results in the usual complex product involving the conjugate of the first entry inner product with the second: $<v|w> = v^* \cdot w$. All the constraining equations for inner product space hold; for instance, number (9) again refers to Section 1.8:

9) Positive definite: $v^* \cdot v$ is greater or equal to 0 and $= 0$ iff $v = 0$.

Also notice that the norm squared of $v = (a + i\ b)$ is again given by just the product: $v^* \cdot v = (a - i\ b)(a + i\ b) = a^2 + b^2 = \|v\|^2$.#

4.2 Complexification

The next method of creating a Hilbert space involves complexification of a Hilbert space, (Halmos, 1958). For a Hilbert space, Hr over the real number field, the procedure results in the complex Hilbert space denoted by Hc. It is of the same dimension as Hr, but it is over the complex number field. In this case, the real scalar multiplication of a vector is extended to the complex scalar multiplication. Among the many procedures for performing complexification, the tensor product method is utilized. The method is described next. Later sections will utilize the complexification of tangent spaces to a fiber when considering quantization on manifolds.

In the MSA global view, sorts similar to those already mentioned for the Hilbert space structure and field structure must be given. These sorts are suggestively defined by C-SCALAR, R-VECTOR, and C-VECTOR. The complex field is denoted by C-SCALAR. The real scalar field exists in complexification, but is not mentionded again. The Hilbert space over the reals is given by R-VECTOR. Similarly, the resulting complex-valued Hilbert space is C-VECTOR. Exactly, the same types of signature sets and corresponding operational names are employed for both Hilbert space specifications; again refer to Section 1.8. However, for complexification, there is an additional signature set with an element of arity two. It contains the binary operator name TENSOR. In this case:

$$\text{TENSOR maps C-SCALAR x R-VECTOR} \rightarrow \text{C-VECTOR}.$$

More details will be given about tensors in subsequent sections, but for the time being this should be sufficient for illustrating the complexification operation. For C-SCALAR, the corresponding carrier sets will always consist of the complex field C. In addition, the actual operation that is named TENSOR will be denoted by $\otimes$, and it is such that $Hc = C \otimes Hr$. Moreover, the operations corresponding to those of V-ADD and S-MULT are defined in an obvious fashion given next.

For any a and b in R and u, v, w, and z in Hr, it follows that:

$$\text{For } V - ADD: (u + iv) + (w + iz) = (u + w) + i(v + z).$$

$$\text{For } S - MULT: (a + ib) \cdot (u + iv) = (a \cdot u - b \cdot v) + i(a \cdot v + b \cdot u).$$

Also, every vector $(u + i\ v)$ in Hc is actually a short notation for $1 \otimes u + i \otimes v$.

The complexification of the inner product follows from the obvious expansion:

$$\text{IN} - \text{PROD:} \quad < u + iv|w + iz > = < u|w > + < v|z > + i(< u|z > - < v|w >)$$

Additionally, the norm squared of u + iv in Hc is $\|u + i\, v\|^2 = \|u\|^2 + \|v\|^2$.

The expression for Hc could have been written in the form Hc = Hr ⊗ C, thus obtaining the same overall result. However, this will not be considered correct, even though they are isomorphic representations. Moreover, most operations in quantum disciplines are non-commutative. For this reason, the polyadic graph description is modified to represent the actual order in which operands are used. This was mentioned earlier for left or right R-modules. Again, for those operators for which the order of operand usage is important, the arrow tails are modified. The corresponding arrow tails have slash (/) indicators denoting the order. Thus, one slash (/) denotes the first operand, two slashes (//) represent the second operand, and so on. For the situation at hand, the arrow for the operator TENSOR has one tail with a single slash emanating from C-SCALAR, and the other tail with two slashes is connected to R-VECTOR. Fig. 4.1 illustrates this fact. In this diagram, all arrows previously utilized in both Hilbert space descriptions and those for a field are omitted. If the other complexification method were used, that is, Hc = Hr ⊗ C, then the slashes in the below figure would be reversed.

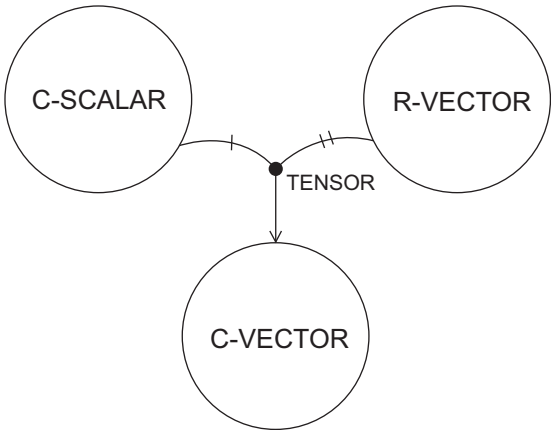


FIGURE 4.1 Polyadic graph for complexification.

Example 4.3:

In this example, we use the complexification just described with carrier set: Hr = R, the real one-dimensional Hilbert space as in Example 4.1. In this case, Hc = C ⊗ R is just the one-dimensional Hilbert space; the vacuum space, Hc = C, is given in Example 4.2. #

At this point, as a reminder, the type of SCALAR determines whether the vector space is real or complex. For instance, if the VECTOR has a carrier set consisting of all complex numbers and the carrier set for SCALAR is the reals, with the usual operations, then the vector space is a real one.

4.3 Dual space used in quantum

The algebraic dual space for a finite-dimensional vector space V over the field of complex numbers C is a set Vd of linear maps (functionals) from V to C. This is often denoted as Hom (V, C). This notation is for the set of multilinear maps from V to the complex field. Besides an algebraic dual space, there is a topological dual space involving bounded linear functionals, and it is described and used in later chapters (Adler, 2015; Mac Land and Birkhoff, 1999).

For the MSA global view of the algebraic dual space, there are two sorts: SCALAR and VECTOR. All the signature sets are as before for both complex field and vector space; however, there exists one additional signature set for the vector space structure. The new signature set itself contains a single set called COVECT. The elements of COVECT are mapping names FUNC1, FUNC2, ... FUNCN. These represent functionals, and N is the dimension of the vector space V. Dual spaces of infinite dimension will be described in later sections. As just mentioned, each element of the set COVECT is a unary linear map name such that:

$$\text{FUNC j: maps VECTOR} \rightarrow \text{SCALAR, for all } j = 1, 2, \ldots, N.$$

So the cardinality of the set COVECT is N. When addition (+) and scalar multiplication ($\cdot$) are defined for the vectors associated with elements within the set COVECT, it becomes a vector space over C. It is of dimension N and is called the dual space of V, and in this document, it is denoted by Vd. Moreover, the following equational identities must hold. Notations to describe these constraints are given by symbols:

SCALAR denoted by a.
VECTOR denoted by v and w.
Functionals in COVECT denoted by f_i and f_j.

The equational identities describing basic vector space operations hold:

1) Vector addition: $(f_i + f_j)(v) = f_i(v) + f_j(v)$.
2) Scalar multiplication: $(a \cdot f_j)(v) = a \cdot (f_j(v))$.
3) Linearity: $f_j(a \cdot v + w) = a \cdot f_j(v) + f_j(w)$. That is, f_j is in Hom (V, C).

All the axioms for a vector space follow from these equational identities. In particular, the zero in Vd is just the zero functional. It maps every element in V to the zero element in C. The global view presented earlier holds true regardless of the field employed; it need not be C. So, for instance, the field could be the reals; in this case, f_j is in Hom (V, R).

Because both vector spaces V and Vd have the same dimension, elements from each are paired in a natural way called a sesquilinear form. This form will result in the inner product and can be justified by the Riesz representation theorem (RRT). However, for finite-dimensional Hilbert spaces like V, it is easy to see that any functional f in Vd can be represented as an inner product. More specifically, there exists a unique vector z in V such that $f(x) = < x | z >$. Moreover, for every vector z in V, there is a corresponding functional, that is, a vector f in Vd such that $f(x) = < x | z >$.

The inner product involving elements of Vd will now be defined. Let g be in Vd and also let the vector w be in V such that RRT is in effect. Then, the inner product in Vd is

given by the identity: $<g|f> = <w|z>$. So the inner product in Vd is inherited from the original inner product in V. As usual, the norm on Vd is induced from the inner product, and the norm is preserved: $\|f\| = \|z\|$. The identification of norms is called an isometry, which in general is a distance-preserving mapping. These results also hold in infinite-dimensional Hilbert space for bounded linear functionals and are of prime importance in quantum disciplines.

The earlier mentioned equational identities and side conditions for an inner product hold with this definition, thus making Vd an inner product space. Because V is finite dimensional, it is a Hilbert space. In infinite dimensions, many of these results are different. The two spaces Vd and V are antiisomorphic; there is a conjugate 1−1, onto, isometric, linear mapping from V to Vd (by associating the vector f with z). Additionally, V is isomorphic to the conjugate space Vd*. In Vd* multiplication of scalar a with vector v is given by $a^* \cdot v$. Fig. 4.2 intuitively illustrates how the dual space Vd is produced by the N functionals FUNC1, ..., FUNCN. A similar procedure and corresponding diagram could be given, showing that U = Hom (V, W) = {f: V → W, such that | f is linear} is a vector space, where W is also a vector space isomorphic to V.

Three distinct vector spaces were described, all of dimension N. It is interesting to relate the basis of Vd with that of the original space V. So let $E = \{e_1, e_2, \ldots, e_N\}$ be a set of basis vectors from V. With abuse of notation, let $F = \{f_1, f_2, \ldots, f_N\}$ be a corresponding basis of Vd. Every covector h in Vd can be expressed in terms of elements of F. In particular, $h = h(e_1) f_1 + h(e_2) f_2 + \ldots + h(e_N) f_N$. Now, a method will be given for finding the actual covectors in F. These vectors correspond to the vectors within the basis E of V. There are numerous bases in a nonzero vector space and several methods for their creation. However, the bi-orthogonality procedure is a standard technique to construct a dual basis in a finite-dimensional vector space (Halmos, 1958).

Both E and F need not be a set of orthonormal or even an orthogonal set of vectors; they just have to be a basis. The procedure for finding the actual covectors in F is first to

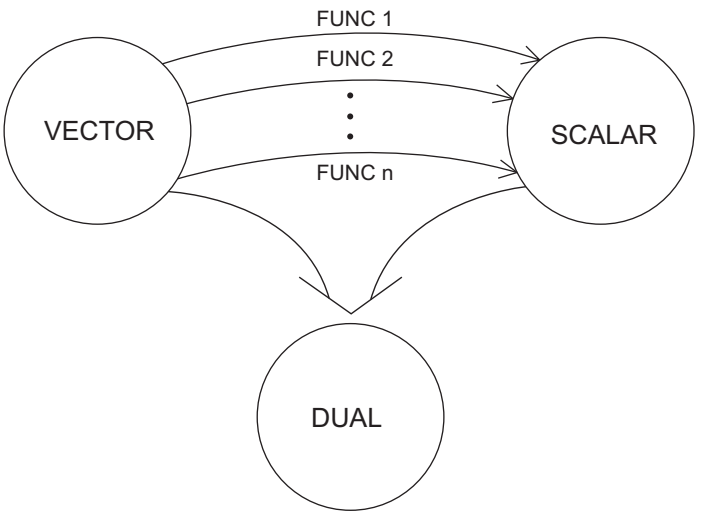


FIGURE 4.2 Polyadic graph for illustrating dual space creation.

realize that each f_j is also a functional. When it is applied to a vector in V, a complex number results. Next, the standard method described here is to take $f_j^*(e_k)$, that is, use the conjugate of f_j and apply it to the basis vector e_k, for all j and k in {1, 2, . . ., N}. This results in a complex scalar. Finally, perform the bi-orthogonality condition: That is, set these complex numbers to zero whenever j is different from k and set it equal to one when j = k. This results in N linear independent equations needed in finding the vectors in F. The resulting basis is called the dual basis of Vd.

A lower MSA view of the dual space will be given along with simple representations of elements from this space. The associated carrier set for COVECT is the set of all functional mappings, and they are often called covectors or one-forms. A simple representation for the vectors in V is to express them as N by 1 complex-valued column vectors. This is useful in determining the covectors in Vd. In this procedure, the first step is to represent in order, each basis element of V as a column vector. Next, represent in order each covector basis element as a 1 by N row vector of conjugated unknown tuples. Simply multiply every row vector (covector = k) by every column vector (vector = j). Each vector multiplication provides a complex number. Doing this for all vectors and covectors results in N equations with N unknowns. Finally, set the resulting equations to zero, except when the order of the indices agrees. In the agreeing situation, that is, when and only when j = k, the result is set to one; otherwise, it is set to zero. This is the bi-orthogonality condition. Solving ends the procedure. An illustration of the method is provided below.

The next example is usually the starting point for studying quantum disciplines. This example involves utilizing kets and bras. These constructs are symbolized by $|>$ and $<|$, respectively. Combining a bra with a ket, in that order, results in the inner product. This is symbolized by $<\|>$, and it is also referred to as a bra-ket. Furthermore, it is common to omit one of the vertical lines in this bra-ket notation. Accordingly, the inner product is denoted by $<|>$, as was illustrated in the previous sections. Also seen before, a comma is sometimes used in the inner product, again replacing these two vertical lines. Both of these notations are standard and will be used throughout this document. In the next example, the underlying Hilbert space is the two-dimensional complex field C^2, along with its dual space. The space C^2 will be seen to be the space of qubits.

Example 4.4:

A basic Hilbert space for quantum technology is the complex Hilbert space C^2, that is, the two-dimensional complex vector space over the complex field. The elements of the carrier set corresponding to VECTOR are called qubits. They will be represented as 2 by 1 column vectors with tuples consisting of complex numbers. Each such column vector is also identified as a ket. The carrier set for SCALAR is again the complex numbers. Specific vector space operations arise from the signature sets. They are the usual 2 by 1 column vector addition and the usual scalar multiplication. This involves complex scalar, c multipliplying a 2 by 1 column vector. A basis for C^2 is very often given by the two qubits: $|0> = (1\ 0)'$ and $|1> = (0\ 1)'$; here (′) denotes transpose, and the result is a basis of column vectors #

Linear combinations, or superposition, of the kets mentioned in the previous example involve coefficients that when absolute squared sum to one. That is, kets of norm one are usually employed, resulting in simple kets. These are more often called pure states as

opposed to mixed states. The convention of calling these vectors pure states will be followed here, but it will be corrected next and again when tensor products of Hilbert spaces are more formally introduced. For the record, states are operators on a Hilbert space rather than vectors within the Hilbert space. Rigorously, a state ρ is a positive linear trace class map, ρ: H $\rightarrow$ H, such that Tr (ρ) = 1. It is called pure whenever there exists a vector v in the Hilbert space H such that ρ_v (u) = [< v, u>/< v, v>] v. So pure states are associated with elements in a Hilbert space. If a state is not pure, then the state is said to be a mixed state. Also, in terms of C* algebras, pure states are unital, positive functionals on a C* algebra. In any case, we will continue with the conventional, operational, and formal notations and descriptions. Concepts such as trace class and positive are made clear in subsequent chapters.

Notice that the Ket $|w> = 1/((2)^{1/2}) \cdot (|0> + i \cdot |1>)$ is an example of another pure state since it has the norm equal to one. It follows from the representation mentioned earlier that $|w>$ can be written as a column vector: $|w> = 1/((2)^{1/2})$ (1 i)′. Observe that ket: $|w>$ has as its second tuple an imaginary scalar value. The set E = {$|0>$, $|w>$} should be considered as an ordered basis because reference to the location of these kets in E will be made. However, it is not an orthogonal basis of C^2. In any case, the corresponding dual space basis F will now be found. The procedure involves using both vectors in E, one at a time. So the Ket $|0>$ will be used as the first element in basis E.

Recall that covectors are represented by row vectors. Their tuples are the complex conjugated values of column vector tuples; they are row vectors. The objective is to find a covector basis F given the basis E = {$|0>$, $|w>$} in V. In order to do this, let the first covector be f = (a* b*). Also, although not used right now, let the other covector be g = (c* d*). Multiply the first covector, that is, the row vector from F with the first column vector from E; doing this gives (a* b*) $\cdot$ (1 0)′ = a*. Using the first vector from each set E, and F, stipulates that the result should be set to one, accordingly a* = 1. Repeat the vector multiplication, by using the first covector from F again, but this time use the second vector from basis E. If this is done, then multiplying the following row vector by the column vector (a* b*) $\cdot$ $1/((2)^{1/2})$ (1 i)′ implies $1/((2)^{1/2})$ (a* + i $\cdot$ b*) = 0. Since a* = 1, we can solve for b* to obtain b* = $-1/i = i$. Therefore, the first covector in the basis F for the dual space Vd is the row vector f = (1 i).

In a similar way, we will now find g, the second covector, (c* d*), in F. This time, (c* d*) $\cdot$ (1 0)′ = 0. This implies that c* = 0. Next, to find d*, set (c* d*) $\cdot$ $1/((2)^{1/2})$ (1 i)′ = 1 / $((2)^{1/2})$ i d* = 1. The number one appeared in the last expression since it involved the second vector in space V and also in Vd. It follows that d* = $-i\,(2)^{1/2}$. Accordingly, g = (0 $-$ i $(2)^{1/2}$). Notice that f and g are not normalized; their two norms squared are $\|f\|^2 = (1\ -i)$ (1 i)′ = 2 and $\|g\|^2 = 2$. However, they still form a basis, F = {f, g}, for the dual space.#

Any covector, h in the dual space, is a unique linear combination of the basis elements in F = {f, g}. An illustration of this linear combination is given next.

Example 4.5:

Given any row vector h in Vd, for instance, say h = (2 1). Then it follows that h has to be represented using the basis for this space, that is, F = {f, g}. This will be illustrated now by using the formula given at the start of this section: h = h(e_1) f + h(e_2) g. In this case, use

$e_1 = |v> = (1\ 0)'$, and $e_2 = |w> = 2^{-1/2}\ (1\ i)'$, which was employed in V previously. Substitute, $f = (1\ i)$, $g = (0\ -i\ 2^{1/2})$, as given earlier. Then, from the formula, $h = [(2\ 1) \cdot (1\ 0)']\ (1\ i) + [(2\ 1) \cdot 2^{-1/2}\ (1\ i)']\ (0\ -i\ 2^{1/2}) = [2]\ (1\ i) + [2^{1/2} + 2^{-1/2}\ i]\ (0\ -i\ 2^{1/2})$. Now multiplying out gives $h = (2\ 2i) + (0 - 2^{-1/2}(2 + i)\ i\ (2)^{1/2}) = (2\ 2i) + (0\ -i\ (2 + i))$, thus obtaining the desired result again, $h = (2\ 1)$.#

The dual space basis just found is of interest only for illustrating its construction along with simple calculations, for instance, using these bases for constructing linear combinations as in Example 4.5. It also provides the background when proving that both spaces V and Vd are of dimension N when this is not assumed. Moreover, it is useful in understanding the double dual space. Otherwise, it is of little use in quantum. This is because the dual basis just found is not normalized. Accordingly, the corresponding bras are not of a unit norm when using this basis. In quantum technologies, the kets and bras are of unit norm. Thus, a more important, but simpler dual basis is given next. It is the basis needed in quantum disciplines.

Consider the orthonormal basis $\{|0>, |1>\}$ of C^2, that is, $E = \{(1\ 0)', (0\ 1)'\}$. The corresponding dual space basis is $F = \{f = (1\ 0), g = (0\ 1)\}$. This is the type of basis for the dual space, which is used throughout quantum computation. Instead of writing this basis as earlier, bras will be used instead. Abusing notation, let the cobasis F be given by $\{<f| = (1\ 0), <g| = (0\ 1)\}$ and $E = \{|0> = (1\ 0)', |1> = (0\ 1)\}'$. An inner product can now be performed, between basis elements from V, using kets, and with those in the dual space using bras. It is easily seen that the bi-orthogonality condition is satisfied. The inner product using bras and kets induces an additional correspondence between the two Hilbert spaces V and Vd.

A natural conjugate linear mapping from the Hilbert space of kets to the Hilbert space of bras is given by K. Here, $K(|v> + |w>) = <v| + <w|$, and $K(a \cdot |v>) = a^* \cdot <v|$, for any complex scalar a, where a^* is the conjugate of a. Also, going the other way, the mapping B from the dual space of bras into the original Hilbert space satisfies $B(<f| + <g|) = |f> + |g>$ and $B(a <f|) = a^* |f>$. Operators B and K are the 1−1, onto, isometric conjugate mappings between V and its dual space. It is an isometric mapping, because norms are preserved. In short, $\| |f>\| = \|<f|\|$. Isometric maps also called isometries are detailed in later chapters.

This result generalizes to any finite-dimensional Hilbert space with kets and bras and in particular to an n quantum-level system, namely n-dimensional Hilbert space C^n. Additionally, it is used in an operational manner with infinite-dimensional topological dual Hilbert spaces, as well as in Banach spaces. Fig. 4.3 illustrates the conjugate linear and isometric mappings using the operations B and K, between V and its dual space Vd.

4.4 Double dual Hilbert space

The double dual space, Vdd of V, is the dual of Vd. Consequently, Vdd = Hom (Vd, C); it is the set of all linear maps from Vd into C. By relating Vd with V, it follows that the two spaces Vdd and Vd are also conjugate linear, 1−1, onto, isometric mapping from Vdd to Vd. It can be said that the double dual Hilbert space Vdd is the same as V (Halmos, 1958). Double conjugating results in the identity map. There is a canonical map from Vdd

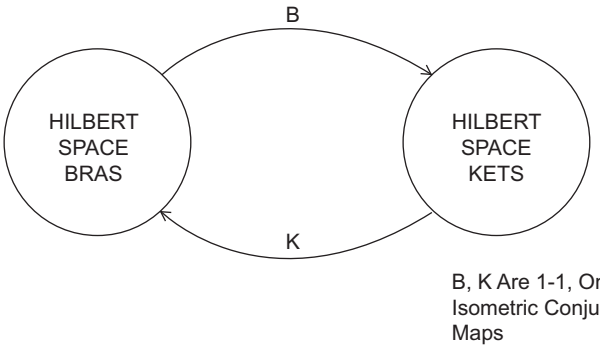


FIGURE 4.3 Graph relating ket and bra Hilbert spaces.

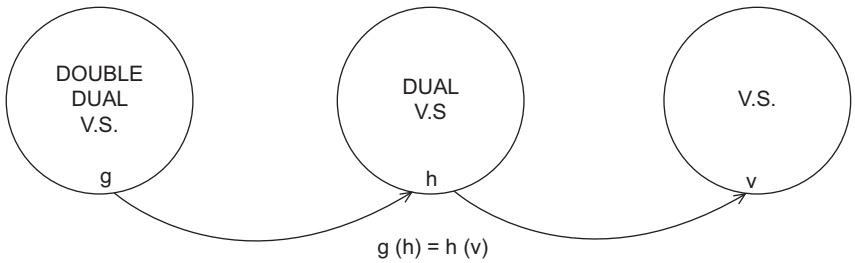


FIGURE 4.4 Canonical isomorphic map for double dual.

to V given in the following paragraph, and it is illustrated in Fig. 4.4. More details are given later.

Let g be in Vdd, that is, g: Vd → C. This is a linear map. It will be shown that g actually sends a vector v in V to C. In the last section, it was seen that when the set of basis vectors $E = \{e_1, e_2, \ldots, e_N\}$ was used for V, then there corresponds a set $F = \{f_1, f_2, \ldots, f_N\}$, which is a basis of covectors in Vd. Of importance is that every covector h in Vd can be expressed as a linear expression using vectors from E as well as basis elements from F. This was illustrated in a two-dimensional vector space in Example 4.5. The general representation is as follows:

$$\text{For h in Vd, } h = h(e_1)f_1 + h(e_2)f_2 + \cdots + h(e_N)f_N.$$

Next, since g is in Vdd, apply g to h in Vd:

$$g(h) = g[h(e_1)f_1 + h(e_2)f_2 + \cdots + h(e_N)f_N].$$

Use the linearity of g and the fact that $h(e_j)$ is a scalar in C; then:

$$g(h) = \big(h(e_1)g(f_1) + h(e_2)g(f_2) + \cdots + h(e_N)g(f_N)\big).$$

Notice that this sum is also in C; that is, it is a sum of scalars. Accordingly, replace each g (fj) by scalar aj and write this sum as follows:

$$g(h) = h(e_1)a_1 + h(e_2)a_2 + \cdots + h(e_N)a_N.$$

Since h is linear, it follows that g (h) can be written as g (h) = h (a_1 e_1 + a_2 e_2 + … + a_N e_N). However, the sum a_1 e_1 + a_2 e_2 + … + a_N e_N is a unique element, v, where v is in V. It follows that by taking any g in Vdd an evaluation of covectors yields a unique element v in V. It is such that g (h) = h (v); this map is illustrated in Fig. 4.4. For every different g in Vdd, there will correspond a different v in V and conversely. This is called the canonical isomorphic map between V and Vdd.

The original space V can be identified with Hom (Vd, C) because by taking any vector v in V we can associate it with a linear evaluation map gv in Vdd. Let gv: Vd → C with gv linear. So, for h in Vd, define gv (h) = h(v), similar to the aforementioned, where g was used. To see that it is linear, use h_1 and h_2 in Vd and a in C; then it follows:

$$gv(h_1 + h_2) = (h_1 + h_2)(v) = h_1(v) + h_2(v) = gv(h_1) + gv(h_2)$$

and

$$gv(ah) = (ah)(v) = a(h)(v) = agv(h).$$

To show an algebraic isomorphism, consider F: Vdd → V, where F (g_1) = v_1, F (g_2) = v_2, and F (g) = v, using h, in Vd.

Then (g_1 + g_2) (h) = g_1 (h) + g_2 (h) = h (v_1) + h (v_2) = h (v_1 + v_2).

Also (a g) (h) = a (g) (h) = a h (v) = h (a v). Natural isomorphism elements of V can be thought of as being linear maps from Vdd into C.

4.5 Outer product

The outer product of vectors and covectors is essential in quantum studies. This operation is often referred to as a pure or simple tensor. In finite dimensions, the result will be a matrix. It will be seen that kets and bras will be useful in describing this operation. First, a rigorous specification for the outer product will be presented in the MSA starting at a global level in a finite-dimensional Hilbert space.

Begin by recalling that the set of all n by n, n = 1, 2, 3, … matrices for fixed n, form an associative unital algebra over the complex field. In this view, the carrier set associated with VECTOR is the set of n by n matrices. All the usual matrix operations of scalar multiplication, matrix multiplication, matrix addition, minus a matrix, partial inverse of a matrix, matrix zero, and matrix identity result from the operational names within signature sets: Correspondingly, the names are S-MULT, BINE, V-ADD, V-MINUS, V-INVERSE, and V-ZERO, and as implied earlier, the identity matrix is obtained from V-ONE. Moreover, all equational identities for a unital associated algebra hold. To describe the outer product, it is useful to define the third sort MATRIX. The associated carrier set is the set of all n by n matrices with all the usual matrix operations resulting from the unital associative algebra construct. Refer to Section 3.1, Fig. 3.1.

The sorts in describing an outer product are VECTOR, SCALAR, and MATRIX. All the signature sets that are applicable to a Hilbert space, complex field, and unital associative algebra structure are assumed to hold along with all their side conditions. However, an

additional signature set containing a single operator name OUTER exists. It is of arity two and such that:

$$\text{OUTER: VECTOR x VECTOR} - - > \text{MATRIX.}$$

Several equational constraints are needed besides those for Hilbert space, the complex field, and the unital associative algebra.

Here, representations for the sorts are as follows:

VECTOR by u, v, and w.
SCALAR by c.
MATRIX by M.

And representative elements of signature sets are given as suggestive symbols:

V-ADD by $+$.
S-MULT by $\cdot$.
OUTER by $\otimes$.

The additional constraints are as follows:

1) Linear: $c \cdot (v \otimes u) = (c \cdot v) \otimes u = v \otimes (c \cdot u)$.
2) Distributive: $(v + w) \otimes u = v \otimes u + w \otimes u$.
3) Distributive: $u \otimes (v + w) = u \otimes v + u \otimes w$.
4) Antisymmetric: $(u \otimes v)'^* = (v \otimes u)$, where the first term $(u \otimes v)$ is transposed and then conjugated.

Fig. 4.5 depicts the outer product operation twice, once in general and then using kets and bras. The sort SCALAR is not illustrated. No other operations are illustrated in this

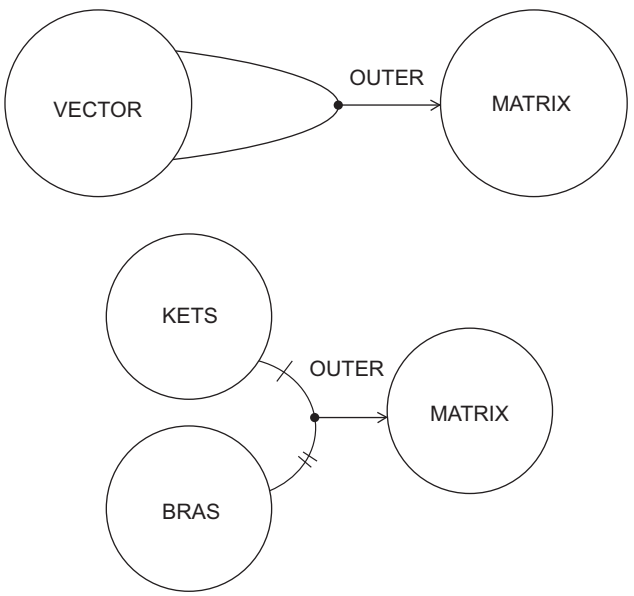


FIGURE 4.5 Polyadic graphs illustrating outer product.

diagram. Operator names solely involving Hilbert space, the complex field, or the unital associative algebra are not shown. In the lower illustration, the polyadic arrow has the first tail corresponding to a ket and the second to a bra. This is the opposite order as used when forming the inner product.

For vectors v and w in C^n, the outer product u is most easily found by forming u = v w′*. So the dimension of v is n by one, and w′* is one by n; therefore, the outer product is an n by n matrix.

Example 4.6:

A simple instance of the outer product can be given with the carrier set associated with VECTOR being C^2. The corresponding carrier set for MATRIX is the unital associative algebra of all 2 by 2 matrices over the complex field. So, for instance, if v = (a b)′ and w = (c d)′ are in C^2, then we can use the operator ⊗, denoting the actual operator whose name is outer product. For chosen v and w, v ⊗ w is the 2 by 2 matrix:

$$|\,ac^* \quad ad^*\,|$$

$$|\,bc^* \quad bd^*\,|$$

As mentioned earlier, the result could have been found using v ⊗ w = v w′*, where w′* is the conjugated transpose of w. So, w′* is a row vector, w′* = (c* d*). However, from the definition of bras, <w| = (c* d*), and kets, |v> = (a b)′, it follows that the representation in Fig. 4.5 holds, that is, v ⊗ w = | v>< w | = (a b)′ (c* d*).#

4.6 Multilinear forms, wedge, and interior products

Before tensors are introduced, the concept of a n linear or multilinear form will be given. For this definition, the cartesian product Π operation is Π_i (Vi) representing the product V1 x V2 x … x Vn. In this case, {Vi} i = 1, …, n is a set of vector spaces over the complex field, and Π represents the cartesian product of all n vector spaces in the set. Then, the operation M: $\prod$(Vi) → C is a multilinear map when M is linear in every coordinate. That is, the equational identities (1) and (2) hold:

1) M (v1, v2, …, a vi, …, vn) = a M (v1, v2, …, vi, …, vn).

2) M (v1, …, vi + v′i, …, vn) = M (v1, …, vi, …, vn) + M (v1, …, v′i, …,vn).
In equation (1) mentioned earlier, vector vi is multiplied by scalar a, and the result is M, where M is the multilinear mapping; similarly, in equation (2) vectors vi and vi′ are added together each is in M, and the result is the sum involving M.
The set of all n multilinear maps M: P (Vi) → C themselves form a vector space V. The multilinear form is called alternating whenever it changes sign by interchanging, permuting, or transposing any two vectors. That is:

3) M (v1, v2, …, vi, …, vj,…, vn) = − M (v1, v2, …, vj,…, vi, …,vn).

The set of all alternating n forms is also a vector space and is represented by AnV*. The dual basis for this space is denoted by $\{e^S$, such that card(S) = n; that is, there are n

elements in S}. Furthermore, S itself is a subset of {1, 2, …, k}, which results in an easy method for keeping track of the basis elements. For k greater than or equal to one, the symbol e^S is defined as $e^S = e^{j1} \wedge e^{j2} \wedge \ldots \wedge e^{jn}$ using S = {j1 <j2 <… <jn}. These jk superscripts are not powers; they are superscripts denoting a dual space basis. The wedge, $\wedge$, in logic denotes the (and) symbol.

For a basis set of V, {ei1, …, ein}, since e^S is the dual space, when e^S operates on vectors in V, scalars are obtained. In this case, e^S(ei1, …, ein) = 1, 0, or −1. This value depends on whether the indexing set for the basis of V, that is, T = {i1 <… <in}, involves the same n, corresponding basis elements as does S. Also the ordering has to be the same. When it does, this yields 1. When the same n elements are employed, but they are permuted, then for an even number of transpositions, 1 is again obtained. For an odd number of transpositions, −1 is the result. Otherwise, the value is zero. That is, it is zero when the dual basis e^S involves different vectors from those in its argument. The argument consists of the vectors in V, namely (ei1, …, ein).

Example 4.7:

Using the aforementioned notation, let k = 3, but n = 2. A basis for the set of all alternating two forms, A2V*, must have two vectors in a basis, but they can be labeled with superscripts from the set, {1, 2, 3}. So, assume that the dual basis in this case is the set $\{e^1, e^3\}$, that is, $e^S = e^{\{1,3\}} = e^1 \wedge e^3$. Using this dual space basis, these vectors e^S can operate on vectors in V, and since they are in the dual space, the result is a scalar. So for the given dual basis, the wedge operating on e2 and e3 results in $(e^1 \wedge e^3)(e2, e3) = 0$. This is because different labeled vectors are employed. Next, $(e^1 \wedge e^3)(e3, e1) = -1$; this occurs because the vectors have the same labels, but in different single-transposed orders. Finally, $(e^1 \wedge e^3)(e1, e3) = 1$, because the vectors have the same labels and identical order.

Next, use linear combinations of vectors from V, with scalars ai and bj in C. The dual operation is applied illustrating the linearity aspects. Let the sum for $v = \sum_{i=1}^{3} ai\ ei$ and also the sum for $w = \sum_{j=1}^{3} bj\ ej$; then $(e^1 \wedge e^3)(v, w) = a1\ b3 - a3\ b1$. This follows, since seven terms among the nine terms have the value zero. For instance, $(e^1 \wedge e^3)(a1e1, b2e2) = a1b2$ $(e^1 \wedge e^3)(e1, e2) = a1b2 \cdot 0 = 0$; this follows first using linearity, that is factoring out a1b2. Then notice that the superscripts are in {1, 3}, but the vectors in V are with labels in {1, 2}.#

For the vector space of alternating n forms, AnV*, its dimension is denoted by dim, and its dimension is calculated using the combination formula of k things then n at a time: (k/n) = k!/(n! (k − n)!). It follows that dim(AkV*) = 1, since in this case n = k, that is, k things taken k at a time, and so (k/k) = k!/(k! (k − k)!) equals one. Moreover, AkV* = {0} for n> k. When n = k, the result is often called a top form, because no larger value of n can be used without obtaining zero. Also A1V* = V*. The degree n, of form w, is denoted by deg(w) and is defined by the relation w is an element of AnV*. Suppose that the wedge or exterior product for w and u are elements of AnV* and AmV*, respectively. Here, w ∧ u is an element of A(n + m)V* defined as follows: (w ∧ u)(v1, …, vn, v(n + 1), …, v(n + m)) = sum over all permutations s of {1, …, n + m}, of the product: [sign(s) w(vs1, …, vsn) u(vs(n + 1), …, vs(n + m)], and preserving the order: s1 <… <sn, and s(n + 1) <… <s (n + m), but also the summation involves shuffling of arguments.

The wedge product is the product in an exterior algebra. It has the following properties (Spivak, 1990):

1) Associative: $u \wedge (w \wedge v) = (u \wedge w) \wedge v$.
2) Homogeneity: $(cu) \wedge v = c\,(u \wedge v) = u \wedge (cv)$, for c a scalar.
3) Distributive: $(u + v) \wedge w = (u \wedge w) + (v \wedge w)$.
4) Distributive: $w \wedge (u + v) = (w \wedge u) + (w \wedge v)$.
5) Anticommutative: $u \wedge v = (-1)^{nm}\,(v \wedge u)$, where u is in AnV* and v is in AmV*.

Example 4.8:

For the case $n = 1$ and $m = 2$, that is, w is in the alternating A1V* space and u is in A2V*, the wedge or exterior product $w \wedge u$ is in A3V*. Its arguments in V are (v1, v2, v3). With w unitary and u binary operators, it follows that all the combinations of $w \wedge u$ are $(w \wedge u)(v1, v2, v3) = w(v1)u(v2,v3) + w(v2)u(v3,v1) + w(v3)u(v1,v2) - w(v1)u(v3,v2) - w(v2)u(v1,v3) - w(v3)u(v2,v1)$. The sign of w(v1)u(v3,v2) is negative because a single transposition of 1, 2, 3 occurs in this element. #

Example 4.9:

For w in $A1R^3$, $\dim(A1R^3) = 3$, since $k = 3$ and $n = 1$, that is, three things taken one at a time are three. In this space, $w = a1e^1 + a2e^2 + a3e^3$ is a row vector, that is, $w = (a1\ a2\ a3)$. More rigorously, there is an isomorphism from this space onto R^3.#

Example 4.10:

This time consider $A2R^3$; again the dimension of this space is three, because three things taken two at a time occur in three ways. However, a typical element of this space is $w = a1(e^2 \wedge e^3) + a2(e^3 \wedge e^1) + a3(e^1 \wedge e^2)$. This can also be identified with (a1 a2, a3). More importantly, when w and u are in $A1R^3$, where w is the sum of bje^j, that is, $w = \sum_{j=1}^{3} bje^j$, and where $u = \sum_{j=1}^{3} cje^j$, then $w \wedge u = (b2c3 - b3c2)\,(e^2 \wedge e^3) + (b3c1 - b1c3)\,(e^3 \wedge e^1) + (b1c2 - c1b2)\,(e^1 \wedge e^2)$. In the last expression that is in $-\ c1\ b2\ (e^1 \wedge e^2)$, this appears because $(b2\ e^2 \wedge c1\ e^1) = b2c1\ (e^2 \wedge e^1) = c1b2\ (e^2 \wedge e^1)$ by linearity and since scalars commute. Finally, $c1b2\ (e^2 \wedge e^1) = -\ c1\ b2\ (e^1 \wedge e^2)$, because $(e^2 \wedge e^1) = (e^1 \wedge e^2)$.

Let w and u be identified as vectors in R^3. In the expression mentioned earlier, observe that the scalar coefficients of the wedge products are the scalar coefficients of the cross product of vectors w and u. This can be seen from the determinant describing the cross product of $w = (b1\ b2\ b3)'$ and $u = (c1\ c2\ c3)'$:

$$\begin{vmatrix} i & j & k \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{vmatrix}$$

$$[b2c3 - b3c2]i + [b3c1 - c3b1]j + [b1c2 - c1b2]k.\#$$

A final note will be given on the wedge product of two vectors is two dimensions. First is that $(w \wedge u)$ is a measure of the noncommutativity of the tensor product of w and u. In this case, $w \wedge u = w \otimes u + u \otimes w$. Accordingly, the wedge product of w and u can be represented as a square matrix with entries $(w \wedge u)_{ij} = (w_i u_j - w_j u_i)$. In three dimensions, the entries of the matrix of $(w \wedge u)$ are zeros on the main diagonal. The other six entries are given by coefficients of the cross product of w, and u, or u and w.

The interior product is like the opposite of the exterior or wedge product. It is associated with a vector field v. The interior product of v is a linear operation; it is denoted by lv and is such that lv: $AnV^* \rightarrow A(n-1)V^*$, where $(lv\ w)(v2, \ldots, vn) = w(v, v2, \ldots, vn)$. This is also called a contraction of w by v. Here, the row vector $(v, \ldots vn)$ is mapped into the function $w(v, v2, \ldots, vn)$, involving n arguments. The convention for $A^{-1}\ V^*$ is to let $A^{-1}\ V^* = \{0\}$. The interior product is nilpotent, that is, two applications yield zero. This implies that it is antisymmetric. The interior product also satisfies a Leibniz-type rule called a derivation. That is, $lv(w \wedge u) = (lv\ w) \wedge u + (-1)^{deg(w)} w \wedge (lv\ u)$.

Example 4.11:

This is an illustration of the Leibniz rule for interior products in R^3. Let a basis set be {e1, e2, e3}, and let v = e1; then $lv(e^1 \wedge e^2 \wedge e^3)$ will be found. Here e^1, e^2, and e^3 are vectors from the dual space, which is a cobasis. First, let $w = e^1 \wedge e^2$ and let $u = e^3$. Note that the degree of w is two and consequently $(-1)^{deg(w)} = 1$. Then, the Leibniz rule says $lv(w \wedge u) = lv\ (e^1 \wedge e^2) \wedge u + 1\ w \wedge (lv\ u) = lv\ (e^1 \wedge e^2) \wedge u$. This is because different basis elements are used in $(lv\ u) = l\ e1\ (e^3) = e^3\ (e1) = 0$. Since the result is $lv(w \wedge u) = lv\ (e^1 \wedge e^2) \wedge u$, another application of the Leibniz rule will be performed. This time let $w' = e^1$ and $u' = e^2$. It follows that $lv\ (e^1 \wedge e^2) \wedge e^3 = [(lvw') \wedge u' - w' \wedge (lv\ u')] \wedge e^3 = 1\ u' \wedge e^3 = e^2 \wedge e^3$. This result follows since deg(w′) = 1, and so a minus sign was used earlier. Also, $lvu' = u'(v) = e^2(e1) = 0$, and finally $lvw' = w'\ (v) = e^1(e1) = 1.\#$

4.7 Many-sorted algebra for tensor vector spaces

Let V be a complex vector space, and now denote its algebraic dual space by V*, as used in Section 4.6. It was symbolized by Vd in Section 4.3. As usual, V* consists of all linear maps f, also known as covectors from V into C. A (p, q) tensor T over the complex field is a multilinear form: T: V* x…x V* x V x…x V → C, where there are p copies of V* and q copies of V in the product space. The quantities p and q are called balances, and their sum is the rank.

The notation given next is the renaming of the sort VECTOR. When using tensors instead of VECTOR, the more common name for the set of all p + q arity tensors is given by a very long representation, it is $V \otimes V \otimes \ldots \otimes V$ x $V^* \otimes V^* \otimes \ldots \otimes V^*$. In any case, the

corresponding carrier set for this sort is {T | T is a (p, q) tensor, and it is also a multilinear form}. The aforementioned name of the set will now be denoted by Y shortened, due to its length. In the name Y of the set, there are p copies of V and q copies of V*. The idea for using this notation Y is that the p elements of V* are applied to the p elements of V. Also, q elements of V are applied to the q elements of V*, resulting in a complex number. Again, this means that when a carrier set is utilized the actual mapping to the scalar field is T: V* x V* x . . . x V* x V x V x. . .x V. → C. The notational difference is used in the MSA to emphasize using x, for the tensor product name and ⊗, as the actual operation.

If this set of tensors is equipped with point-wise addition, (+), and scalar multiplication, (.), then the set becomes a vector space over the complex field. All the operator names and equational identities for a complex field remain as before. However, an additional signature set is needed in this MSA description for the vector space structure. The new signature set has an element of arity 1 and is named TEN, corresponding to the actual multilinear operator T, representing a tensor. All other elements of the signature sets for the vector space structure hold, with some names more suggestively defined. Elements of the signature sets that are relevant are given next below. In the following: T-ADD denotes tensor addition; ST-MULT denotes tensor mutiplication by a scalar; TEN denotes tensor multiplication.

T-ADD: Y x Y → Y. Note that this addition is defined for tensors of fixed balance.

$$\mathrm{ST-MULT:\ C-SCALAR\ x\ Y \rightarrow Y}$$

$$\mathrm{TEN:\ Y \rightarrow C-SCALAR}$$

$$\mathrm{T-MINUS:\ Y \rightarrow Y}$$

$$\mathrm{T-ZERO,\ Is\ in\ Y}$$

The polyadic graph given in Fig. 4.6 illustrates these five operations. No operations solely involving the complex field structure are illustrated. With these operations, the arity sequence associated with a tensor vector space is given by (1, 2(1, 1), 2(1, 1)). This follows because there are two different signature sets each containing unary operator names, and two distinct signature sets each containing binary operator names, as well as a zero-ary operator name.

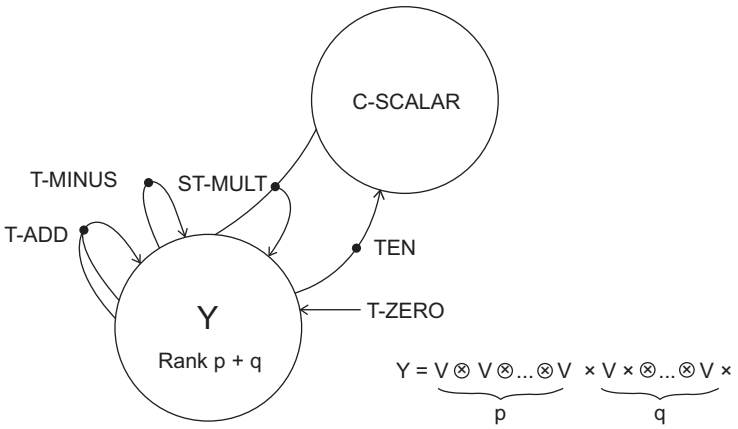


FIGURE 4.6 Tensor vector space.

As previously mentioned in the MSA low view, the carrier set consists of all the (p, q), tensors T. They will now correspondingly be denoted as Tpq. In addition, the vector space itself will be denoted by Vpq.

Example 4.12:

The vector space V00 is the field of scalars; in this case, it is the complex field C.
The vector space V01 = V* = {f: V → C}, so the elements of this space are covectors. #

Example 4.13:

In a finite-dimensional vector space, it follows that V10 is the vector space of all vectors, that is, V10 = V. This follows from the isomorphism between V and V**. That is, this space consists of all linear maps from V* into C, and therefore, it is V. The vector space V11 = {T11: V* x V → C} = End (V*). The actual tensors are T11 = {v ⊗ f, such that v is in V and f is in V*}.#

The expression End (V*) is the set of endomorphisms on V*. In a finite-dimensional vector space, it is also true that End (V*) = End (V)′, where the prime indicates that the endomorphism algebras are the same except for transpose. So, column vectors in V correspond to row vectors in V*. For matrix operators M and N, note that the quantity (M N)* = N* M*. Also if M: V → C, such that M v = w, then v′ M′ = w′ and so M′: V* → C.

Example 4.14:

Examples of tensors, f and g with balance (0, 2), are often considered an inner product. For f and g in V*, f ⊗ g: V x V → C. So for v and w in V, f ⊗ g (v, w) = f(v) • g (w). The latter product is like a dot product. Additionally, this product need not be positive definite.

Also, this tensor is a metric tensor, which is defined on manifolds or surfaces. It enables distances and angles to be defined on these structures. (Lee, 1997). Metric tensors are such that:

1) Symmetric: g (u, v) = g (v, u).
2) Positive semidefinite: g (u, u)> 0 and g (u, u) = 0 iff u = 0.

Metric tensors are a field of tensors defined on a tangent space manifold. These tensors form a tensor bundle.#

A tensor with balance (0, N), with a vector space of dimension N, is known as an N form. It is also called a top form, a volume form, or a determinant. This is described in the next section.

4.8 The determinant

The determinant is defined for endomorphisms on an N-dimensional vector space (Lang, 2002). In order to obtain a determinant, a top form is needed and it is defined below. First, an n form, f, is a T0N tensor that is totally antisymmetric. Here, n is an

integer such that it is in [0, N]. If n = 0, then a T00 tensor is obtained, which is a scalar. Otherwise, for any transposition of the elements vi and vj, the sign (plus or minus) of f (v1, v2, . . ., vn) changes. For an even number of transpositions, a plus 1 is used to multiply f; otherwise −1 is employed. If k forms are considered with k> N, they are zero. So nontrivial k forms exist for k less than or equal to N. When k = N, the top-degree form is obtained.

The top form is a T0n form, where n = N. When this holds, there are nonvanishing f and g, both top forms; then there is a scalar c such that g = c f. This becomes an equivalence class. The choice of one such top form, f, is called the volume on V0N. A vector space with a chosen top form is called a vector space with volume. When v1, v2, . . ., vN are N vectors in V0N, it is said that the volume spanned by these vectors is by definition vol (v1, . . ., vN) = f (v1, . . ., vN).

The determinant is taken of the endomorphism, A, and results in a scalar. It utilizes a basis of V0N, say e1, e2, . . . eN, along with some volume form f on V0N. It is defined by det A = f (A (e1), A (e2), . . ., A (eN))/ f (e1, e2, . . ., eN). (Lee, 1997; Sakai, 1995)

Example 4.15:

Consider the endomorphism A: $C^2 \rightarrow C^2$, defined using the diagonal matrix D, with a nonzero scalar c, followed by 1 on the main diagonal. So D =

$$|c \quad 0|$$

$$|0 \quad 1|$$

For any v = (a b)′ and w = (d e)′, in C^2 and scalar α, then D preserves scalar multiplication and vector addition. That is,

$$D[\alpha v + w] = D(\alpha a + d \quad \alpha b + e)' = (c\alpha a + cd \quad \alpha b + e)' = \alpha(c\ a \quad b)' + (c\ d\ e)' = \alpha Dv + Dw.$$

So, this is a vector space endomorphism. The determinant provides a result invariant under a change of basis. In this case, for a unitary matrix U, when it is employed and it is such that U U* = U*U = I, then the determinant of D = c = det(U D U*). #

4.9 Tensor algebra

The tensor sum can be found only for tensors with the same balance. A tensor product, on the other hand, can be applied to arbitrary tensors. There are four sorts in the MSA description in this case, and they are SCALAR, VECTOR1, VECTOR2, and VECTOR12. The first sort denotes the scalars in the complex field as before. The next three sorts again denote vectors in vector spaces; in this case, vectors are actually tensors. The tensor product will utilize tensors from tensor spaces VECTOR1 and VECTOR2, resulting in tensors in VECTOR12.

Besides all the previous signature sets for vector spaces and fields, there is an additional signature set. It has a single operator name, which is actually a tensor product; it is T-MULT and is of arity two. Fig. 4.7 illustrates this operation. The carrier set for SCALAR

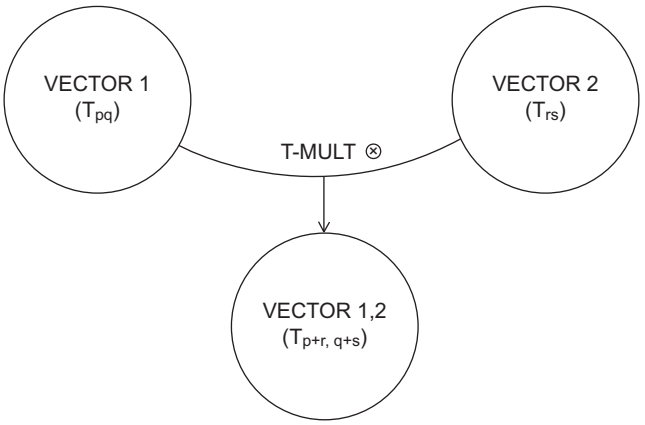


FIGURE 4.7 Operations involving tensors.

consists of all the complex numbers. The carrier set for VECTOR1 is the set of all Tpq vectors in Vpq, and the carrier set for VECTOR2 is the set of all tensors Trs in Vrs. The vector space of all these tensors is an unital associated, commutative algebra. In this case, the V-ADD operation is for the addition of two tensors both with exactly the same balance.

The actual tensor product, T-MULT, is denoted ⊗. Here, Tpq ⊗ Trs is a (p + r, q + s) tensor. The polyadic graph in Fig. 4.7 illustrates these operations on tensors.

(Tpq ⊗ Trs) (v1, v2,. . .,vp, . . .vp + r, f1, f2,. . .fq, . . .fq + s) = Tpq (v1, . . ., vp, f1, . . ., fjq) x Trs (vp + 1, . . ., vp + r, fq + 1, . . ., fq + s), where the multiplication is in the complex field. This shows that the tensor product operation is commutative. Not shown in this diagram is the complex field structure, as well as all the vector space and tensor operations already given in Figs. 1.2 and 4.6.

Example 4.16:

Consider the two (1, 1) tensors S and T over the complex field. Also, let f be a covector in V* and let v be a vector in V. Also assume that c is a complex value. Then V ⊗ V* = {T, such that T:V* x V → C}, and it follows that V ⊗ V* is a vector space:

$$(S + T)(f, v) = S(f, v) + T(f, v),$$

$$(cT)(f, v) = cT(f, v)$$

All the equational identities hold, in particular, T = 0 is the zero, that is, it is the T-ZERO or the V-ZERO of the vector space.#

In Vpq, where the dimension is finite say N, then let e1, e2, . . ., eN be the basis of V = V10 and let f1, f2, ..., fN be the dual basis found previously on V* = V01. Recall that this covector dual basis is obtained from the bi-orthogonality condition. So the tensor can be written Tpq (f1, f2, . . . fp, e1, e2, . . . eq). Also, the tensor Tpq can be reconstructed from its components, using r = e1 ⊗ e2 ⊗ . . . ⊗ep ⊗ f1 ⊗ f2 ⊗ . . . ⊗ fj. More details, as well as examples of tensor operations, are given in the next section.

4.10 Many-sorted algebra for tensor product of Hilbert spaces

An MSA description for the tensor product of Hilbert space H1 and Hilbert space H2 is given next from a high view. It is also a Hilbert space. The sorts are presented in an obvious fashion; they are SCALAR, VECTOR1, VECTOR2, and VECTOR12. The signature sets are exactly as in that of Hilbert space, except that an additional signature set is needed to provide the tensor multiplication. This signature set has a binary operator name called T-MULT. It is such that:

T-MULT: VECTOR1 x VECTOR2 → VECTOR12.
As in previous MSA high-level presentations, denote:
V-ADD by +
S-MULT by x
T-MULT by ⊗
SCALAR by c
VECTOR1 by v1, w1
VECTOR2 by v2, w2

Moreover, all the equational identities for Hilbert space hold true in this structure along with the additional equational constraints:

1) Linearity: c x (v1 ⊗ v2) = (c x v1) ⊗ v2 = v1 ⊗ (c x v2).
2) Distributive: (v1 + w1) ⊗ w2 = v1 ⊗ w2 + w1 ⊗ v2.
3) Distributive: v1 ⊗ (v2 + w2) = v1 ⊗ v2 + v1 ⊗ w2.
4) Inner Product: <v1 ⊗ v2, w1 ⊗ w2> = < v1, w1> <v2, w2>.

With these constraints, VECTOR12 becomes a Hilbert space designated by H12. The polyadic graph in Fig. 4.8 illustrates the operation T-MULT in producing VECTOR12. Again, the sort SCALAR is not shown, nor is any operational name involving Hilbert spaces, except for T-MULT. Moreover, sorts VECTOR1 and VECTOR 2 are denoted by H1 and H2, respectively.

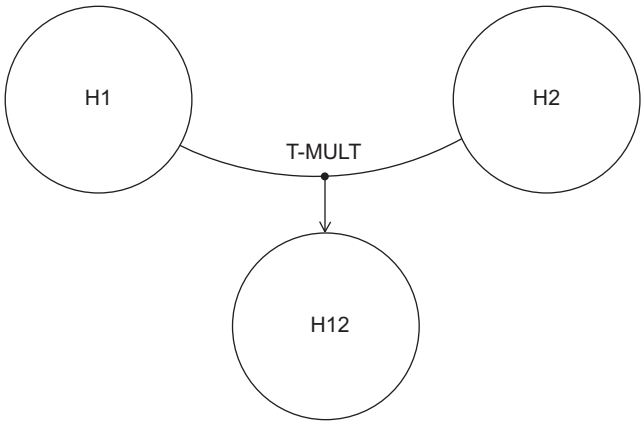


FIGURE 4.8 Tensor product of Hilbert spaces.

Preparing for simple examples of tensor Hilbert spaces, consider the Hilbert space $H1 = H2 = C^2 = \{|v>\}$, that is, the set of all kets. So, the carrier set here is the set of all 2 by 1 complex-valued column vectors. In C^2, the most general normalized single qubit in this space is given by the ket: $|v> = e^{iq}$ [cos (t/2)|0> + e^{ip} sin (t/2)|1>], p, q in [0, 2 π), t in [0, π]. In C^2, the two-by-one vector, for the ket, |v> is given by |v> = (e^{iq} cos (t/2) e^{iq} e^{ip} sin (t/2))′.

As stated earlier, note that this ket does have norm one. Using the inner product shows that the norm squared is $\| |v> \|^2 = \|e^{iq}\|^2$ [(cos (t/2)^2 + (sin (t/2))^2] = 1. All other kets in this space are equal to the linear combination of two distinct kets |u> and |v> of this form. For a pure element, the coefficients of the resulting ket must sum to one using absolute squared values of the coefficients. For instance, let |w> = a |u> + b |v> with a and b in C; then |a |^2 + |b |^2 = 1. These elements appear on the surface of the very important Bloch sphere, described in the next chapter. One of the most instructive low views utilizing H1 ⊗ H2 is given in the next example.

Example 4.17:

Let the carrier set for VECTOR12 be $H1 \otimes H2 = C^2 \otimes C^2 = C^4$, which is the space of two qubits and of four-dimensional complex-valued column vectors. In H1, use the basis, $v1 = 1/2^{1/2}$ (|0> + |1>) = $1/2^{1/2}$ (1 1)′, and w1 = |0> = (1 0)′. Also in H2, use the basis, v2 = 3/5 |0> −4i/5 |1> = (3/5 −4i/5)′, and w2 = i |1> = (0 i)′. The tensor product of individual basis elements from H1 and H2 will be found. Thus, there will be four such products. For simplity representation in C^4 will be employed. In this case, if v = (a b)′ and w = (c d)′ in C^2, then the tensor product u of v and w, is u = (ac* ad* bc* bd*)′ in C^4. The star is the complex conjugate.

The results become understandable when represented, in terms of C^4. An easy to use basis in C^4 is |0 0> = (1 0 0 0)′, |0 1> = (0 1 0 0)′, |1 0> = (0 0 1 0)′, and |1 1> = (0 0 0 1)′. Thus, T-MULT(v1, v2) = v1 ⊗ v2 = $1/2^{1/2}$ (3/5 4i/5 3/5 4i/5)′. Also, w1 ⊗ w2 = (0 −i 0 0)′. Additionally, v1 ⊗ w2 = $-i/2^{1/2}$ (0 1 0 1)′ and w1 ⊗ w2 = (3/5 4i/5 0 0)′. All the previous results could have been found directly using kets. Notice that $\|$v1 ⊗ v2$\|$ = 1/2 [(3/5)^2 + |4i/5 |^2 + (3/5)^2 + |4i/5|^2] = 1. Similarly, $\|$w1 ⊗ w2$\|$ = 1, and the same is true for all other tensor products specified earlier. #

Example 4.18:

Let the carrier set of VECTOR be the set of all complex-valued 2 by 1 column vectors, this is again $H1 = C^2$. Tensors involving H1 ⊗ H1 can also be easily found. The tensor product of |u> = (1/ $2^{1/2}$) (|0> + |1>), and |v> = (1/ $2^{1/2}$) (|0> − |1>), will be determined. It is |u> ⊗ |v> = 1/2 (|0> ⊗ |0> − |0> ⊗ |1> + (|1> ⊗ |0> − |1> ⊗ |1>). This can be written: w = 1/2 (|0, 0> − |0, 1> + |1, 0> − |1, 1>). Note that the resulting tensor product is also of norm one. This can be seen directly by using bras and kets and observing when they are orthogonal and normalized. So, <w,w> = 1/4 <<0, 0 | − <0, 1 | + <1, 0 | − <1, 1 |, |0, 0> − |0, 1> + |1, 0> − |1, 1>>. This could be expanded into sixteen inner products. Each of the sixteen inner products would then be calculated using the equational

identity number four for tensor Hilbert spaces. 4) Inner Product: $<v1 \otimes v2, w1 \otimes w2> = < v1, w1> <v2, w2>$.

To see this in depth by using the inner product mentioned earlier, begin with the first tuple in the first bra and form the inner product of it with each of the four first tuples within the kets. The results are $<0, 0> = 1$, $<0, 0> = 1$, $<0, 1> = 0$, and finally, $<0, 1> = 0$. Next form the inner product of the second tuple in the first bra with each of the four second tuples within the kets. This gives $<0, 0> = 1$, $<0, 1> = 0$, $<0, 0> = 1$, and $<0, 1> = 0$. Keeping track of these two results and then multiplying these entries in corresponding order gives all zeros except for one term, $<0|0><0|0> = 1$. That is, the other three products are zero. Repeating the process for the other bras will yield a single value 1 in each product, when multiplying in order. So, $<w,w> = 1/4\ (<0|0><0|0> + <0|0><1|1> + <1|1><0|0> + <1|1><1|1>) = 1$.#

4.11 Hilbert space of rays

The next two examples of a Hilbert space include the concept of rays for states. This is first described using projective Hilbert space concepts. The second example utilizes ordered fields and convex cones. Both examples also assume that the states are pure states.

Example 4.19:

Consider a fixed complex Hilbert space H. The projection on H results in the subset M of H, M = {v, w, nonzero, such that v = a w for some nonzero constant, a in C}. This defines a ray as an equivalence class. This structure satisfies the reflexive, symmetric, transitive (RST) equivalence relations. To see that this is a relation, assume that v, w, and z are all nonzero in M; also, let a and b be nonzero in C. The following constraints do hold:

Reflexive: v = 1 v.
Symmetric: v = a w, implies w = 1/a v, since a is nonzero.
Transitive: v = a w, w = b z, implies v = a b z; accordingly, all RST relations hold.

Vector zero is not in this structure, and consequently, M is not a subvector space. It is not even an additive groupoid. This follows since if both v = a w and v = −a w are added together then the zero vector is obtained. There may be many subsets like M, and these constitute the rays in a Hilbert space H.

Each of these rays can be thought to be comprised of two infinite segments of a straight line. The line passes through the zero in H and does not contain it, but it does contain all other points on the line. In particular, if $H = C^2$ then the projection in this case is the complex projection line, also known as the Bloch sphere, described in the next chapter. Hopf fibration, also described later, provides the underlying theory for conclusions relating to the Bloch sphere and is mentioned in Chapter 10 on Fiber bundles.#

Generalizing the result from the previous example to an n quantum-level system results in equivalence classes and can be thought of as complex lines through, but not including the origin in C^n. The lines form a complex projective space, $CP^{(n-1)}$, which is

$(C^n - \{0\})$ / (v ~ av), for a in C, nonzero. There is a one-to-one correspondence between $CP^{(n-1)}$ and n-dimensional quantum systems. A single qubit is in CP^1. A more in-depth presentation of projective spaces is provided in the next example. It will involve showing that the outer product of a simple vector with itself is positive semidefinite or otherwise called nonnegative definite.

Example 4.20:

In this example, consider Hilbert space, Hr, with the carrier set for sort SCALAR being the real field, R. The real field is an ordered field. Moreover, for any two elements, a and b, in R, one of the three cases holds. Either a <b or b <a or a = b is true. Additionally, the ordering involving the relation, less than or equal to, is reflexive and transitive. Moreover, it is antisymmetric, that is, if a is less than or equal to b and b is less than or equal to a, then a = b. Also, the defining equational inequalities for R to be an ordered field involve the constraints:

1) Additive: if $a < b$, then $a + c < b + c$.
2) Multiplicative: If 0 <a and 0 <b, then 0 <a • b.

Recall that the outer product of two vectors results in a matrix. Now, we show the trace class operator, density function, or pure state; |v> <v| is a real-valued nonnegative definite matrix. This will be illustrated by showing that the associated quadratic form of the matrix is nonnegative definite.

So consider the qubit |v> = (a b)′ in C^2 then |v> <v | = (a b)′ (a* b*) = M, the quadratic form Q = (x y) M (x y)′ = (a a* x + b a* y a b* x + b b* y) (x y)′ = a a* x^2 + 2 Re (a b*) x y + b b* y^2, which is greater or equal to zero. This shows that the matrix M is positive semidefinite. M = | v> <v | =

$$\begin{vmatrix} aa^* & ab^* \\ ba^* & bb^* \end{vmatrix}$$

Finally, M is also a ray in the convex cone in Hr. A convex set is such that the line segment containing any two points within the set is also fully contained within the set. A cone is a subset of an ordered field, Hr. For any element P in the subset and scalar a, where a is nonnegative in R, then a P is in the subset. A convex cone is a cone where a P + b Q is in the cone for any nonnegative a, b in R, and any P and Q in the cone.#

4.12 Projective space

For a Hilbert space H, over the field C, an associated projective space PH can be defined. It is the set of all rays L, in H where PH = {L, such that L is a complex one-dimensional subspace of H}. Consider H − {0}, with its induced topology from H, and let T: (H − {0}) → PH. Then, choosing a single nonzero point on each ray L shows that T is an onto map. Additionally, choosing the final topology for PH with respect to T makes PH a topological space. Here, in the final topology, open sets U in PH are those for which T^{-1}

(U) are open in H − {0}. For v, w in H − {0}, an equivalence relation can be defined by v ∼ w means v = aw for a, nonzero in C. Equivalence classes were illustrated in Example 4.19. Denote the quotient space, (H − {0})/∼ by H∼, and define the quotient map T∼: (H − {0}) → H∼. This produces a unique homeomorphism M, M: H∼ → PH. Elements in H − {0} are often referred to as state vectors, whereas elements L of PH are called the rays. When there exists a unit norm, they are referred to as pure states.

The projective Hilbert space inherits an inner product from H. In this case, the binary function IN: PH x PH → [0, 1], and for v, w in H − {0}, then IN(v,w) = | < v,w > | / (‖v‖‖w|)|.

Example 4.21:

When a quantum system is prepared to be in state v in PH for measurement and the state of the observable is w in PH, then the probability of the event is interpreted to be IN $(v,w)^2$.#

When the dimension of H is greater than one, the PH is a complete semi-metric space with semi-metric given by d: PH x PH→ [0, infinity], where for L, K in PH, the distance d (L,K) = inf ‖v − w‖, where v is in L, w is in K, and ‖v‖ = ‖w‖ = 1.

Example 4.22:

If H = C^2, the projective space PH is the set of all straight lines in the complex plane C, which go through the origin. In particular, if L is the y-axis and K is the x-axis, then d(L,K) = $2^{1/2}$. This follows, since unit lengths are employed, and the distance from point (1, 0) to point (0, 1) is $2^{1/2}$.#

The Fubini-Study metric D is defined on a projective Hilbert space (Sakai, 1995). It is an equivalent metric to d, in the previous example. The Fubini-Study metric is such that D: PH x PH → [0, infinity], where D(L,K) = arccos(|L,K|). Equivalency is shown by the inequalities: For all L and K in PH, d(L,K) is less than or equal to D(L,K) that is less than or equal to $2^{1/2}$ d(L,K).

Isometries can be defined in projective spaces. If PH1 and PH2 are projective spaces, then A: PH1 → PH2 is called an isometry whenever it is distance preserving, | < A(L), A (K) > | = | < L, K > |, for all L and K in PH1. An isometric automorphism is a mapping A, where A: PH → PH when A is a 1−1, onto isometry. This is also called a Wigner automorphism or symmetry of the quantum system (Chevalier, 2007). There are many types of isometries such as translations, rotations, and reflections as well as glide reflections, which are reflections about a line followed by translations. These symmetries form a group. Distinct isometries are described as being linear whenever A obeys:

1) Additive: A (L + K) = A (L) + A (K).
2) Homogeneous: A(aL) = a A(L), for a in C.

However, if (1) holds and

2′) Conjugate Homogeneous: A(aL) = a* A(L),

then the isometry is said to be conjugate linear. A theorem of Wigner's; involves isometries in a Hilbert space of dimension greater than one For an isometry A: PH → PH, there exists a linear or conjugate linear isometry S such that S: H → H. Moreover, A = S′, which is an isometry on PH mapping the ray cv into the ray cSv, for v in H − {0}.

It is important to mention a related topic, namely antiunitary or conjugate linear unitary operations. In particular, for a bijection T: H → H satisfying the three conditions below, the operator is called an antiunitary map; here a and b are in C and v and w are in H:

1) Adjoint Inverse: T T* = T* T = I.
2) Additive: T(v + w) = T(v) + T(w).
3) Conjugate Homogeneous: T(a v) = a* T(v).
In terms of inner products, this is often condensed into:
4) Inner product, <T(v), T(w)> = < v, w>*.

However, the last representation is useful, but care must be taken because all operators in this document are assumed to be linear, unless specified otherwise.

Example 4.23:

If T is antilinear, then so also is T^{-1}. For a non-zero, $T^{-1}(av) = w$, implies that $av = T(w)$, or $v = 1/a\ T(w) = T(1/a^* w)$. From this $T^{-1}(v) = 1/a^* w$, or $w = a^* T^{-1}(v)$, thus showing the inverse is also antilinear. It also follows that T^{-1} (i) $T = -i\ T^{-1}T = -i$.#

References

Adler, J., 2015. Linear Algebra Done Right. Springer, 978-3-319-11079-0.
Chevalier, C., 2007. Wigners theorem and its generalization. Handbook of Quantum Logic and Quantum Structures. Elsevier, p. 429.
Halmos, P., 1958. 0-387-90093-4 Finite Dimensional Vector Spaces, 1st ed. Van Nostrand.
Lang, S., 2002. 978-0-387-95385-4 Algebra, Graduate Texts in Mathematics, vol 211. Springer-Verlag.
Lee, J., 1997. 978-0387-98322-6 Riemannian Manifolds. Springer Verlag.
Mac Land, S., Birkhoff, G., 1999. Algebra. AMS Chelsea, 0-8218-1646-2.
Sakai, T., 1995. Mathematical Monographs No. 149 Riemannian Geometry. American Mathematics Society.
Spivak, M., 1990. A Comprehensive Introduction to Differential Geometry. Publish or Perish, Inc., Houston.

This page intentionally left blank

CHAPTER

5

Quantum and machine learning applications involving matrices

5.1 Matrix operations

The polyadic graph in Fig. 5.1 provides a global view of several operational names involving the sort MATRIX. These names are in the context of an n by n complex-valued matrix, M. The sorts in this diagram consist of n-dimensional complex Hilbert spaces of KETS, as well as the same or dual space of BRAS, along with the complex SCALAR field. To keep the diagram noncluttered, all operational names previously mentioned for Hilbert space as well as for the complex field are omitted in this diagram. Also, omitted are all the operator names used to define the unital associative algebra structure for MATRIX as detailed in Fig. 3.1.

The arity sequence for the MATRIX structure in the present case is (0, 7 (4, 3), 0,1) and can be correlated with Fig. 3.1. There are seven unitary operational names and one trinary

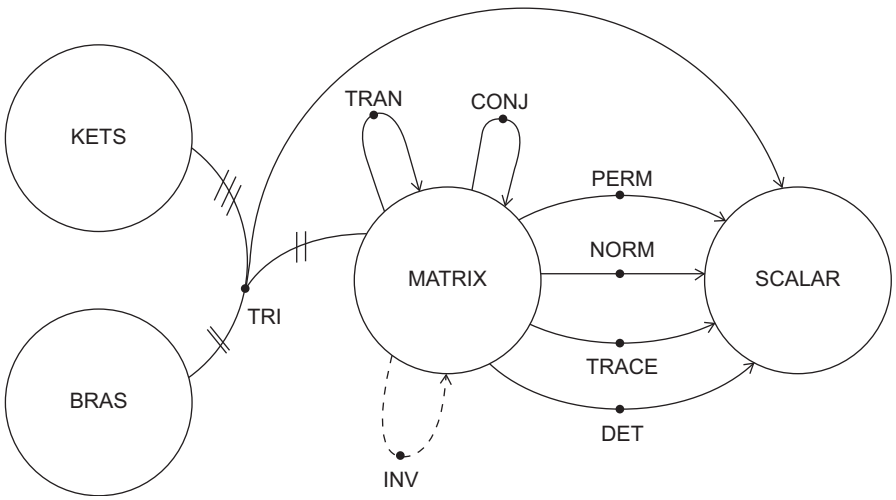


FIGURE 5.1 Polyadic graph involving matrix operations.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00008-4

operational name. This is the first instance of a trinary operator name. The truth is that many other operations could be given using arity three operations. For instance, even one of the most basic structures in quantum technology can be defined this way. That is, an abelian group can be defined utilizing a single trinary operation; in this case, it is often called a heap (Hollings and Lawson, 2017).

For the MATRIX structure, the names of all the operators within their signature sets are given next. We begin with:

$$\text{Trinary Opertation: TRI: BRA} \times \text{MATRIX} \times \text{KET} \rightarrow \text{SCALAR}.$$

This is a conjugate trilinear form. In the complex case, TRI is linear in its last two arguments only. Of special importance in the real case is when the bra and ket are equal. The interpretation in this case is to use v such that $<v| = |v>^*$. As a result, a quadratic form is obtained, $<v|M|v>$. In any event, however, notice in the diagram that the tails of the corresponding arrow are marked with slashes corresponding to the order of operands for TRI. The order is as follows: first a BRA, then MATRIX, and finally KET.

The next four operational names all belong to the same signature set; they are described in the following paragraphs.

$$\text{Unary Operation: DET: MATRIX} \rightarrow \text{SCALAR}.$$

The determinant was introduced earlier involving vector space endomorphisms in Section 4.8. It will be mentioned again in subsequent chapters. Endomorphism methods produce component-free evaluation formulas for the determinant. However, in a less abstract setting, the determinant is also called an (unique up to scalar multiplication) alternating multilinear functional. A determinant is very important in Fock spaces. Here, the Slater determinant is employed in providing fermion count status as well as entanglement existence.

Next, another important matrix operation corresponds to the trace operational name; it is such that:

$$\text{Unary Operation: TRACE: MATRIX} \rightarrow \text{SCALAR}.$$

The trace will be given from a more global view in subsequent sections. It involves contractions of tensor products. The trace of a matrix plays a crucial role in employing statistical and probabilistic methods in quantum areas. Averages are based on using the trace of the density matrix. Illustrations and examples are provided in subsequent chapters.

The permanent operational name is such that:

$$\text{Unary Operation: PERM: MATRIX} \rightarrow \text{SCALAR}.$$

The permanent is similar to the determinant, but it does not use the one and minus one alternating signs in the Leibniz rule for the calculation of a determinant. It keeps all signs equal to positive one. The permanent is also employed in Fock space for calculation of the number of bosons.

The norm has operational name given by:

$$\text{Unary Operation: NORM: MATRIX} \rightarrow \text{SCALAR}.$$

Here, NORM stands for the norm of a matrix. It is a global term. However, among the many distinct norms that exist for a matrix, only the Hilbert-Schmidt or Frobenius norm will be mentioned in the following. In this case, $\|M\|$ equals the square root of the sum of all the matrix elements, each absolute value squared. This norm was chosen because it can be used in illustrating a Banach algebra with carrier set MATRIX.

The final three operator names all are unary and all belong to the same signature set. They are the following:

TRAN, CONJ, and INV they all map: MATRIX: → MATRIX.

TRAN is the transpose of a matrix, CONJ is the conjugate transpose, and INV is the partial inverse whose true domain is those matrices whose determinant is nonzero. Notice the dotted arrow in Fig. 5.1. It would be a solid arrow if the sort MATRIX denoted elements within the general linear group GL (n, C) or GL (n, R) of n by n matrices whose determinants are nonzero.

It should be pointed out that the term transpose (TRAN) only applies to matrices. In general, the adjoint operation takes the place of transpose for linear operators. These facts will be mentioned again later. The adjoint will perform similarly to the transpose. For a matrix M, where M: $C^n \rightarrow C^m$ and M is an m row by ncolumn matrix, then the inner product is used for v in C^m and w in C^n as $<v,Mw> = v^*Mw = (M'\ v^c)'\ w = <M^*v,w>$. The star (*) is the conjugate transpose. The exponent (^{c}) is the conjugate. When the real scalar field is used instead of the complex field, the conjugate of any real number is itself and (*) = (′).

The next example illustrates the use of the transpose involving complex quantities. In this example, (^{c}) refers to the conjugate operation.

Example 5.1:

Consider the carrier sets for VECTOR to be C^2 and C^3, and where, n = 3 and m = 2. Let the carrier set for MATRIX be all 2 by 3 complex-valued matrices, and let M: $C^3 \rightarrow C^2$, where M =

$$\begin{matrix} |0 & i & 2| \\ |1 & 0 & 0| \end{matrix}$$

For v in C^2, where v = (2i −1)′, and w in C^3, w = (3i −2i 1)′, then <v, M w> = <(2i −1)′, (4 3i)′> = (−2i −1) (4 3i)′ = −11i = v*Mw. The quantity $M'v^c = (-1\ 2\ -4i)'$, and accordingly, $(M'v^c)' = (-1\ \ 2\ -4i)$ and $(M'v^c)'w = (-1\ \ 2\ \ -4i)\ \ (3i\ \ -2i\ \ 1)' = -11i$. Finally, $<(M'v^c)^C$, w> = <M*v, w> = <((−1 2 4i)′, (3i −2i 1)′> = −11i.#

5.2 Qubits and their matrix representations

As previously mentioned, a ket followed, and multiplied, by a bra is employed in denoting outer products. For instance, using M, a two by two complex-valued matrix and with notational abuse: M = |v><w|. The following example will illustrate the use of the outer product in

quantum. In particular, it will be observed that a normalized vector ρ in a Hilbert space is written as $|v><v| = \rho$. This quantity is often referred to as a trace class operator, a density function, or even a pure state. More generally, the outer product is used as an orthogonal projection operator. Here a ket $|v>$ of norm one projects onto the subspace spanned by $|v>$ using $|v><v|$. Illustration of this important operation is given in the MSA polyadic graph in Fig. 5.2. A special notation is used here. A dot on each tail of an arrow indicates that the operands are identical elements of the chosen sort or isomorphic sort. Identical elements for the situation at hand mean a $|v> = a^*<v|$. The operator K in the diagram is the 1−1, onto, isometric, conjugate operation illustrated in Fig. 4.2. The ket followed by bra operation is motivated in the next example. In this case, the carrier set for VECTOR corresponds to two-dimensional complex-valued vectors with the usual operations. As a consequence, the carrier set for sort MATRIX is the set of all 2 by 2 complex-valued matrices.

Example 5.2:

Refer to Example 4.6 where the fundamental Hilbert space setting is C^2. That is, the carrier set for VECTOR is the set of all 2 by 1 complex column vectors identified with kets. The outer product of the two kets $|r> = (a\ b)'$ with $|s> = (c\ d)'$ is a 2 by 2 matrix E. Entries of E are the following: $E11 = ac^*$, $E12 = ad^*$, $E21 = bc^*$, and $E22 = bd^*$. The same result could have been found by multiplying the 2 by 1 column vector

$|r> = (a\ b)'$ by the 1 by 2 row vector $<s| = (c^*\ d^*)$, corresponding to a bra representation for $|s>$. Again, $|r><s| =$

$$\begin{matrix} |ac^* & ad^*| \\ |bc^* & bd^*|. \end{matrix}$$

Consider any three vectors u, v, and w in C^2. The inner product $<w, v>$ times the ket u equals the outer product $|u> \otimes w>)|$ times ket v. That is, $<w|v>|u> = (|u> \otimes w>)|v>$. This is the usual definition of an outer product as a rank one operator. This

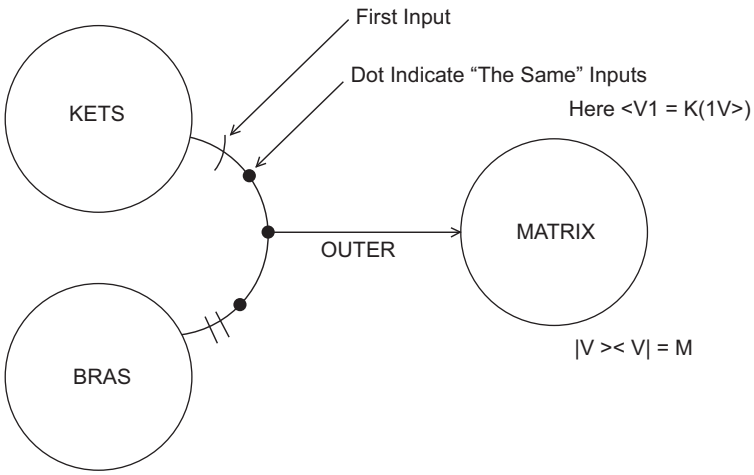


FIGURE 5.2 Illustration of identical operands in polyadic graph.

identity will now be illustrated: for instance, let u = (a b)′, v = (e f)′, and w = (c d)′. Then forming the left hand side gives the 2 by 1 column vector, $<w|v>|u> = ((c^*e + d^*f)\ a\ (c^*e + d^*f)\ b)'$.

Next, forming the right hand side $(|u>\otimes|w>)|v>$, it is given below by the two by two matrix $(|u>\otimes|w>)$, followed by the column vector (e f)′,

$$\begin{matrix} |ac^* & ad^*| & |e| \\ |bc^* & bd^*| & |f|. \end{matrix}$$

Multiplying the 2 by 2 matrix times the 2 by 1 column vector gives a new 2 by 1 vector:

$$\begin{matrix} |(ac^*\ e + ad^*\ f)| \\ |(bc^*\ e + bd^*\ f)|. \end{matrix}$$

From this, the identity $(|u>\otimes|w>)|v> = <w|v>|u>$ is seen to hold.#

More generally, any column vector (n by 1) multiplying a row vector (1 by n), in that order, corresponds to the outer product. When using bras and kets, the conjugate must be employed in the row representation of a bra. In any case, the results of the multiplication are an n by n matrix. Arbitrary ket-bra multiplications each of the same degree provide outer products in an n-dimensional Hilbert space.

The following examples continue with Hilbert space C^2, but this time the origin is not included because states can never be zero, and arbitrary phase factors are ignored. In quantum areas, overall phase differences cannot be detected. Numerous representations and lower level views of simple (pure) states within the MSA are provided next. As far as quantum applications are concerned, the following representation is most important; it is that of Poincare or Bloch sphere. See Fig. 5.3. It generalizes to Lie groups. In particular, the special unitary group, SU(2,C), is a 2 by 2 matrix representation for manipulating qubits. This involves the Pauli basis as well as the commutator operation and the antisymmetric Levi-Civita symbol. Additionally, in the orthogonal group O(3, R), the pure state is given by a 3 x 1 column vector with an invariant unit norm. High views of the algebraic and topological foundations of Lie groups and Lie algebras are provided in a forthcoming chapter.

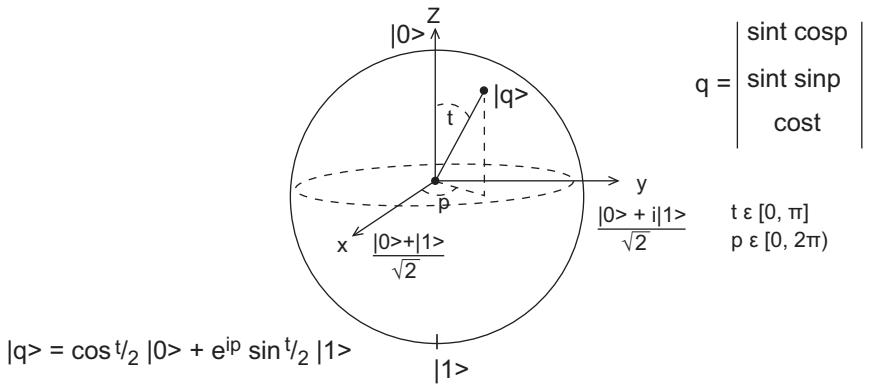


FIGURE 5.3 Bloch sphere.

Example 5.3:

With the exception of arbitrary phase, the most general single qubit is described in this example. It is the 2 by 1 vector: $|q> = (\cos(t/2)\, e^{ip} \sin(t\,2))'$, where p is a real number in $[0, 2\pi)$ and t is real valued in $[0, \pi]$. Correspondingly, it can be written using a linear combination of kets: $|q> = (\cos(t/2)|0> + e^{ip} \sin(t\,2)|1>$. Making use of the outer product or projection operation: $|q><q|$, this results in the 2 by 2 self-adjoint matrix. It is positive definite, of trace one, with the nonnegative determinant. It is a representation of $|q>$, called the density matrix, $\rho =$

$$\begin{vmatrix} \cos^2(t/2) & \sin(t/2)\cos(t/2)e^{-ip} \\ \sin(t/2)\cos(t/2)e^{ip} & \sin^2(t/2) \end{vmatrix}.\#$$

The aforementioned example provided a Hilbert space representation of a pure state, a single qubit. Previously mentioned, the most general qubit was illustrated; it is given by $e^{is}\,|q>$, where s is a real number in $[0, 2\pi)$ and $|q>$ is given earlier. A most intuitive, as well as a most important, representation of a single qubit is given next. It ignores the global phase e^{is}, but involves the Bloch sphere and is described next.

A point on the surface of the Bloch sphere is always a pure state. The point is given by the 3 by 1 vector, q, in the Lie group O(3). It is obtained using spherical coordinates or using normalized Stokes parameters needed in polarization identification. This representation for the single qubit is called the spherical representation of the qubit and is the 3 by 1 column vector: $q = (\sin(t) \cos(p) \sin(t) \sin(p) \cos(t))'$. Here, t denotes an (elevation, latitude type) angle in $[0, \pi]$, and p is a (longitude type) angle in $[0, 2\pi)$. Of special importance are the north and south poles. These occur by using $t = 0$ and $t = \pi$, $p = 0$, respectively, thus obtaining kets $|0>$ and $|1>$, respectively. See Fig. 5.3. So, for instance, using $|q> = (\cos(t/2)\, e^{ip} \sin(t\,2))'$, with $t = 0$, gives $|0> = (\cos(0/2)\, e^{ip} \sin(0\,2))' = (1\ 0)'$ in C^2. Additionally, the spherical representation of this qubit is $q = (\sin(0) \cos(p) \sin(0) \sin(p) \cos(0))' = (0\ 0\ 1)'$. In three dimensions, this is $x = 0$, $y = 0$, and $z = 1$; it represents the north pole. A similar representation can be given for $|1>$. In this case, the representation on the Bloch sphere is $(0\ 0{-}1)$, and this is the south pole.

The next representation involves the three Pauli matrices: s1, s2, and s3. Along with the 2 by 2 identity matrix, these four matrices form an orthogonal basis for all 2 by 2 complex self-adjoint matrix Hilbert space. So, for any self-adjoint matrix T, that is, $T = T^*$, it follows that $T = n_1\, I + n_x\, s1 + n_y\, s2 + n_z\, s3$, where all the n_j are real numbers.

The Pauli matrices are given below in order, s1, s2, and s3:

$$\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & -i \\ i & 0 \end{vmatrix} \quad \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$$

The self-adjoint matrix T mentioned earlier, when the Pauli matrices are substituted, is given by:

$$\begin{vmatrix} n_1 + n_z & n_x - in_y \\ n_x + in_y & n_1 - n_z \end{vmatrix}.$$

Example 5.4:

The general Pauli matrix T, given above will be used as a density matrix ρ. Since the trace of ρ equals one, it follows that trace(ρ) = $n_1 + n_z + n_1 - n_z = 2\ n_1$; this implies $n_1 = 1/2$. Accordingly, making the substitution, $\rho = 1/2\ I + n_x\ s1 + n_y\ s2 + n_z\ s3$. Let r be the following column vector, that is, r = (2 n_x 2 n_y 2 n_z)′. Also use s1, s2, and s3 in making the Pauli column, 3 by 1, vector, s = (s1 s2 s3)′. This quantity is really a 6 by 2 matrix. It is of these dimensions when the Pauli matrices are actually substituted into the column vector s. Another important representation is $\rho = (I + <r, s>)/2$; it is a condensed version of the representation of ρ mentioned earlier. Since ρ is positive semidefinite, its eigenvalues are nonnegative and so also is its determinant. Here, Det (ρ) = $n_1^2 - n_z^2 - n_x^2 - n_y^2$ is greater than or equal to zero. Therefore, 1/4 is greater than or equal to $n_z^2 + n_x^2 + n_y^2$. This implies that $|r|^2$ is less than or equal to one. When $|r| = 1$, a pure state is obtained and is a point on the surface of the Bloch sphere. When $|r| < 1$, this indicates a mixed state and is a point in the interior of the Bloch sphere. Example 5.8 illustrates a maximally linear mixed state at the center of the Bloch sphere.

Example 5.5:

This example will illustrate the effect that an application of the s2 Pauli matrix has on a typical qubit, |q> = (cos (t/2) e^{ip} sin (t/2))′. This qubit is located on the surface of the Bloch sphere, because it has norm one. The first step is to form the matrix s2 times this vector: s2 |q> = s2 (cos (t/2) e^{ip} sin (t/2))′ = ($-ie^{ip}$ sin (t/2) icos (t/2))′. Overall phase does not matter, so factoring out $-ie^{ip}$ gives s2 |q> = (sin (t/2) $-e^{-ip}$ cos (t/2))′ when this global phase is dropped, that is, it is ignored. This quantity can be written as (cos ((t − π)/2) $-e^{-ip}$ sin ((t + π)/2))′ = (cos ((t − π)/2) e^{-ip} sin ((t + 3π)/2))′ = (cos ((t − π)/2) e^{-ip} sin ((t − π)/2))′.

The aforementioned calculation shows from an application of s2, p becomes minus p, and t becomes t − π. On the Bloch sphere, the original qubit |q> is transformed into a new qubit s2|q> whose new location has a longitude change from angle p to -p; additionally, a latitude change of 180° minus its original angle. Thus, for an application of s2 on a given qubit, there is a change in hemisphere from the original hemisphere occupied by the qubit. For instance, the north pole is transformed into s2 |0> = (0 i)′, which is the south pole. Factoring out the imaginary number renders s2 |0> = i (0 1). Ignoring global phase gives the new value of t = π and p = 0. As another instance, consider the original point being the south pole. So, s2 |1> = (-i 0)′. Converting this to -i(1 0), and then remembering that global phase is not relevant, gives s2|1> = |0>.#

As in the aforementioned example, all the Pauli matrices act in a similar fashion involving the three spatial directions. All the Pauli matrices given earlier are both Hermitian symmetric, that is, they are self-adjoint as well as being unitary matrices. The latter statement means that multiplying one of these matrices on either side by its conjugate transpose results in the identity matrix.

Example 5.6:

An illustration will be provided, showing that the second Pauli matrix s2 is self-adjoint as well as being a unitary matrix, that is, s2 = s2 * and s2 · s2*′ = I. Note that s2 is given in order, below followed by s2* and s2*′, where * denotes conjugation and the prime indicates matrix transpose.

$$\begin{matrix} |0 & -i| & |0 & i| & |0 & -i| \\ |i & 0| & |-i & 0| & |i & 0| \end{matrix}$$

A simple matrix multiplication verifies the claim that s2 · s2*′ = s2*′ s2 = I.#

The last representation for q that will be given now is the 2 by 2 matrix, M. It is an element in the SU(2, C) Lie group. The matrix M involves the three by one, spherical vector q and the three by one, Pauli vector s, given earlier. In this case, it is the inner product of two vectors. So, <q, s> = M = sin (t) cos (p) s1 + sin (t) sin (p) s2 + cos (t) s3. The entries sk in the aforementioned results are all 2 by 2 Pauli matrices; they multiply the sinusoidal. Multiplying the sinusoids with the substituted 2 by 2, Pauli matrices, when everything is added together, results in the 2 by 2 matrix M =

$$\begin{matrix} |\cos(t) & e^{-ip}\sin(t)| \\ |e^{ip}\sin(t) & -\cos(t)\ |. \end{matrix}$$

Pauli matrices are often defined slightly different from the definitions given earlier. It should be mentioned, as a first instance of this is in their corresponding Lie algebra, su(2, C). In this case, the Pauli matrices are normalized: that is, Sj = sj/2 for j = 1, 2, 3. Also, in this algebra, the commutator operator is used, not regular matrix multiplication. The commutator operation is [A B] = A B-B A. Matrix multiplication actually is not defined in the Lie algebra; it is a metaoperator needed for obtaining the commutation operation.

Example 5.7:

As an application of the commutator operation, consider the Pauli matrices: [Si, Sj] = Si Sj − Sj Si = i E_{ijk} Sk. Here, the structure constants, also called the antisymmetric Levi-Civita symbol E_{ijk}, ijk were employed. The first i in front of the i E_{ijk} symbol is the square root of minus one. The symbol E_{ijk} is equal to one or minus one depending on whether the ijk pattern undergoes an even number of transpositions or an odd number, respectively. Plus one is used for an even number of transpositions, and minus one is used otherwise. A transposition is the interchanging of two entries involving the ordering of the numerals: 1, 2, and 3.

For instance, $[S_3\ S_2] = S_3\ S_2 - S_2\ S_3 = -iS_1$. First of all, the minus sign occurs since there are an odd number of transpositions: 1 2 3 say became 3 2 1; using a single transposition, interchange 1 and 3. The solution involves S_1, since the commutator employed S_2 and S_3. The product of 4 S_3 S_2 and 4 S_2 S_3, as well as 4 (S_3 S_2-S_2 S_3), is illustrated as follows, in the order specified:

$$\begin{matrix} |0 & -i| & |0 & i| & |0 & -2i| \\ |-i & 0| & |i & 0| & |-2i & 0|. \end{matrix}$$

Multiplying the last matrix by 1/4 gives $-iS_1$.#

Mixed states are the convex combinations of pure states and other mixed states. The coefficients in the linear combination are always nonnegative and sum to one in this case. These coefficients are interpreted to be probabilities. On the other hand, pure states have probabilities being the absolute squared scalar coefficient. Let $|u>$ and $|v>$ be pure states. Additionally, assume that c and d are greater than or equal to zero. Then $|z> = c|u> + d|v>$, where $c + d = 1$, is called a mixed state, and c and d can be considered to be probabilities. The resulting mixed state ket is a convex combination of other kets in this case. Elements like these appear inside, interior to the Bloch sphere. Recall that the Bloch sphere is a representation of the projected space, $CP^1 = (C^2 - \{0\})/(z \sim az)$, where a is nonzero in C. The interior is sometimes called the Bloch ball. Thus, mixed states are in the Bloch ball. The origin in the Bloch ball has the state corresponding to one half of the identity. It is called the maximal linear state. The state carries no information about a qubit.

Example 5.8:

As mentioned earlier, a most simple and important example of a mixed state is the maximal linear state, $|v> = 1/2|0> + 1/2\ |1> = 1/2$ I. As far as the Bloch ball is concerned, it was indicated earlier that $|v>$ is located at the center of this ball. In this case, $r = 0$, because $\rho = |v><v| = 1/2$ I and $\rho = (I + < r, s>)/2$, from Example 5.4.

5.3 Complex representation for the Bloch sphere

A pure qubit can be represented with u, and v in C as $|q> = u|0> + v|1>$, where $|u|^2 + |v|^2 = 1$. This representation is sometimes used instead of the more popular representation given in the previous sections: $|q> = (\cos(t/2)|0> + e^{ip} \sin(t/2)\ |1>$, p in $[0, 2\pi)$, and t in $[0, \pi]$. Here, the global phase, ϕ in $[0, 2\pi)$, is ignored, so the full description of a qubit is $e^{i\phi}\ |q>$. In the Bloch sphere representation, $e^{i\phi}\ |q>$ is mapped onto a single point on the surface of the sphere S^2 in R^3. The phase ϕ does not show up on the Bloch sphere. The coordinates of this point are given by the spherical coordinate vector described before. Now it is additionally described in terms of the complex coefficients, u and v,

$$X = \sin(t)\cos(p) = 2\mathrm{Re}(u^*v)$$
$$Y = \sin(t)\sin(p) = 2\mathrm{Im}(u^*v)$$
$$Z = \cos(t) = |u|^2 - |v|^2$$

By identifying u with $\cos(t/2)$ and v with $e^{ip} \sin(t/2)$ provides justification for the X, Y, and Z identities. The density matrix using the kets given earlier is $\rho = |q><q| = (u|0> + v|1>)^*(u|0> + v|1>) = |u|^2|0><0| + |v|^2|1><1| + u^*v|0><1| + v^*u|1><0|$. Accordingly, the density matrix $\rho =$

$$\begin{matrix} ||u|^2 & u^*v| \\ |v^*u & |v|^2|. \end{matrix}$$

The off diagonal entries of this matrix are often called coherences. The process of utilizing transformations to remove these entries is called de-coherence.

Letting $u = x + iy$ and $v = z + it$, with x, y, z, and t real-valued, then the ball S^3 is defined as follows: $x^2 + y^2 + z^2 + t^2 = 1$. Equivalently, S^3 can also be defined using two complex numbers, that is, use u and v in $|u|^2 + |v|^2 = 1$. This shows that the single qubit Hilbert space is S^3. In this space, $e^{i\phi}\ |q>$ is the equation of a circle with parameter ϕ in [0, 2π). The Bloch sphere is identified with the projective space CP^1 or S^2.

Finally, using the outer product or projection operation: $|q><q|$, this results in the 2 by 2 matrix representation of the pure state density matrix, $\rho =$

$$\begin{matrix} |\cos^2(t/2) & \sin(t/2)\cos(t/2)e^{-ip}| \\ |\sin(t/2)\cos(t/2)e^{ip} & \sin^2(t/2)\ |. \end{matrix}$$

The bock vector can be written as a density matrix; that is, ρ can also be given again in terms of spherical coordinates as $2\ \rho =$

$$\begin{matrix} |1 + Z & X - iY| \\ |X + iY & 1 - Z\ |. \end{matrix}$$

As before, justification amounts to substitution. For instance, for the matrix entry ρ, using $Z = \cos(t)$, $1 + Z = 1 + \cos(t) = \cos^2(t/2)/2$.

Example 5.9:

The eigenvalues for the Pauli matrix s1 are easily found since the characteristic equation is $\lambda^2 - 2\ \lambda$ trace (s1) + det (s1) $= \lambda^2 - 1 = 0$. Consequently, the eigenvalues are $+1$ and -1. The eigenstates or eigenvectors for Pauli s1 matrix are $1/2^{1/2}(|0> + -|1>)$. The corresponding density matrices are given by $2\rho =$

$$\begin{matrix} |1 & + -1| \\ |+ -1 & 1| \end{matrix}$$

5.4 Interior, exterior, and Lie derivatives

This section is a continuation of Section 4.6. The principal content of this section is a description of the Lie derivative. First, however, the interior derivative will be reviewed. Not previously mentioned, the interior derivative is always specified relative to a vector field X. For any n form w, $n > 0$, $(l_X\ w)\ (X2, X3, \ldots Xn) = w(X, X2, \ldots Xn)$. This operation acts on a sequence of vector fields, X2, X3, ...Xn. This is the same as the interior product. The result is an $n - 1$ form. For a function, that is a zero form, the interior derivative is zero.

Example 5.10:

Let w be a two form over the dual space of R^2, where $w = e^1 \wedge e^2$, and let $X = e_2$. The interior product $(l_X w)\ (v) = w\ (X, v) = (e^1 \wedge e^2)\ (e_2, v) = e^1\ (e_2)e^2\ (v) - e^1\ (v)\ e^2\ (e_2) = 0 - e^1\ (v)\ 1 = -e^1\ (v)$. So notice that the result of applying the interior derivative to a two form resulted in a one form.#

Notice that the exterior product, which is the wedge product described in the previous chapter, Section 4.6, is different from the exterior derivative, which is just the d operation. The exterior derivative d, of an n form w, increases the degree by one. For the two forms, v and u of degrees n and m, respectively, the exterior derivative of $w = v \wedge u$ is $dw = (dv) \wedge u + (-1)^n v \wedge d(u)$. However, there is a close relationship between the interior and exterior derivative, and this relationship is described next.

The duality between the exterior and the interior derivatives is given by the Cartan magic formula $L_X w$, where $L_X w = d(l_X w) + l_X dw$ (Zubelevich, 2012). In this formula, X is a vector field, l_X is the interior derivative, L_X is the Lie derivative, and d is the exterior derivative. Finally, for w, an n form, Cartan's formula could be written as $L_X w = \{d, l_X\} w$. The brackets {A, B} are the Poincare brackets. Recall that these brackets are such that $\{A, B\} = A B + B A$. This magic formula is often used as the definition for the Lie derivative $L_X w$, on the space of differential forms.

For a function f with respect to a vector field X, the Lie derivative is the directional derivative, $L_X f = d f \cdot X = X [f]$. The Lie derivative of vector fields, Y along X, is such that $L_Y X = [X, Y]$, since [X, Y][f] = X [Y [f]]-Y [X [f]]. Additionally, it is the contraction of the exterior derivative of f with X, that is, $L_X w = I_X dw + d(I_X w)$, which again is the Cartan formula. Note that the Lie derivative also obeys Leibnitz rule: For $w = v \wedge u$ and X a vector field, $L_X w = (L_X v) \wedge u + v \wedge (L_X u)$.

Lie derivatives do not change the degree of the forms they act on. Another way of keeping the degree of the form invariant is to use both the interior and exterior operators since the former lowers the degree, while the latter raises the degree. The order of operation matters, and in any case, Cartan magic formula is an illustration of keeping the degree of the form invariant (Marsden, 2003).#

For regular functions f, that is, 0 forms, both the exterior and Lie derivatives agree. However, the meaning of derivative is somewhat different for these two operators. A Lie derivative provides a rate of flow or change in a zero form, whereas the exterior derivative can be viewed as the inverse operation of integration (Yano, 1957).

5.5 Spectra for matrices and Frobenius covariant matrices

In finite-dimensional Hilbert spaces H, the spectral characteristics of operators T are realized by matrices M. In particular, the spectral decomposition will be described for matrices with distinct eigenvalues. It involves a superposition of pairwise orthogonal projections onto the eigenspace Va, for eigenvalue a. So, Va = {a such that T v = a v, v nonzero in H}.

Let Pa be an orthogonal projection onto Va. When a1, a2,. . ., an are the distinct eigenvalues of T, the spectral decomposition of T is $T = a1\ Pa1 + a2\ Pa2 + \ldots + an\ Pn$. Also, $T^2 = a1^2\ Pa1 + a2^2\ Pa2 + \ldots + an^2\ Pan$. This follows since Paj and Pak are orthogonal and projections are idempotent. Moreover, for any polynomial involving T, $p(T) = p(a1)\ Pa1 + p(a2)\ Pa2 + \ldots + p(an)\ Pan$. The functional calculus, indicated in later chapters, extends the decomposition from polynomials to continuous functions f. One of the most important formulas for functions of a matrix is given next.

The Lagrange-Sylvester interpolation formula is described for a matrix A, over the complex field. It is assumed that A is a k by k matrix with distinct eigenvalues. The objective is to find the function f of a matrix A, that is, f (A). The value f (A) is the sum $\sum_{i=1}^{k}$ f(ai)Ai. In this sum, there appears the product of f (ai), for distinct eigenvalues ai of A. This quantity multiplies the Frobenius covariants Ai (Horn and Johnson, 1991). Each of the k covariants Ai is found as a product j = 1 to k but j not equal to i of the quotient (A − aj I)/(ai − aj).

Example 5.11:

As an instance of this formula, consider k = 2 by 2 matrix A,

$$\begin{vmatrix} 2 & 1 \\ 3 & 4 \end{vmatrix}$$

The characteristic equation is $(2-\lambda)\ (4-\lambda)-3=\lambda^2-6\ \lambda\ +\ 5=0$. Then, the distinct eigenvalues are a1 = 5 and a2 = 1. In this order, we will find the Frobenius covariants A1 and A2. Since k = 2, the product will only have one term for each Frobenius covariant. Thus for A1, j only will equal 2, and therefore A1 = (1/4)(A − I). In a similar fashion, for A2, j only equals 1, and consequently, A2 = −(1/4) (A − 5I). These Frobenius covariants need to be calculated only once. From this, functions of A are found using f (A) = f (5) A1 + f (1) A2. This can be done for f (A) equal to e^A, sin (A), cosh (A), and so on. As a simple check, f (A) = A will first be found. It is 5 A1−1 A2. The corresponding Frobenius covariants are 4 A1 and 4 A2 given as follows in that order:

$$\begin{vmatrix} 1 & 1 \\ 3 & 3 \end{vmatrix} \quad \begin{vmatrix} -3 & 1 \\ 3 & -1 \end{vmatrix}$$

Multiplying the first matrix above by 5/4 and the second matrix above by −1/4 and adding them together sums to A.

As another simple example, A^2 will be found. It is 25 A1−1 A2 =

$$(25/4)\begin{vmatrix} 1 & 1 \\ 3 & 3 \end{vmatrix} - (1/4)\begin{vmatrix} -3 & 1 \\ 3 & -1 \end{vmatrix} = \begin{vmatrix} 7 & 6 \\ 18 & 19 \end{vmatrix}.\#$$

A similar formula involving the Frobenius covariants exists for the situation where there exists repeated eigenvalues (Horn and Johnson, 1991).

5.6 Principal component analysis

The Karhunen-Loeve transform (KLT), the Hotelling transform, and the principal component analysis (PCA) are different names for essentially the same transform method. In machine learning, the terminology is usually referred to as the PCA. It is a linear feature reduction technique, and it is often called an unsupervised decomposition algorithm. Being unsupervised, like the k means method illustrated in Example 2.1, Section 2.1, suggests that

it needs an exhausting amount of computation in all applications. This is true. In general, PCA is a method for finding the vectors of greatest influence and then reducing the dimension of the data set by removing data related to less influential vectors. PCA is called an eigenproblem. The actual solution is centered about finding eigenvectors v_j and their eigenvalues λ_j. All eigenvectors with the largest absolute value eigenvalues λ_j are usually retained. Then, the method is to reduce the dimensionality of data by removing eigenvectors possessing small absolute valued eigenvalues. Actually, this is a form of data compression, also called source coding. From a statistical point of view, the PCA method is a method for determining the direction that maximizes the variance or correlation of the data.

The PCA was invented by Pearson (1901). Early on, the Hotelling transform was used in probability and statistics applications (Hotelling, 1933). Additionally, along with stochastic processes, the KLT (Loeve, 1977) found applications in engineering applications.

The reason for using eigenvectors is because they provide directions of principle components. Principal components are always orthogonal to each other, and intuitively they form orthogonal axes. The eigenvalues represent the units of spread captured by each principal components. The PCA does not just discard original data; it maps the data into a new feature space. It is in this space that the reduction is performed. As mentioned before, PCA is mainly used in dimension reduction, but it is sometimes used for data visualization as well as feature extraction. The algorithm helps identify variables that are dependent and remove them. That is, multicollinearity is identified and dealt with using PCA. PCA reduces the dimension set, but does not reduce much of the information content when components are discarded.

PCA is also utilized for noise reduction, as well as anomaly detection, and as an encryption device. The last application manifests itself by utilizing the eigenvector method given herein, but retaining all the eigenvectors, that is, do not discard any eigenvectors. It acts much like a hashing code. Additionally, as mentioned earlier, the PCA is used for removing random noise. In this situation, the PCA is performed, and then the inverse PCA is created. When applied, the inverse result will often yield better information content than the original data set. Anomaly detection is another useful application of the PCA. It again involves inverting the PCA. Once this is performed, a comparison is made in determining the difference between the original and re-constructed data. Anomalies are detected when this difference is larger than it should be. That is, the difference is larger than the difference utilizing previous comparisons when no anomalies exist. The learning aspect here is still unsupervised; there is no need to label the data.

Arbitrary data sets can be reduced using PCA, but prenormalization of all data within the original data is paramount. Often data sets are converted to have mean zero. Every column of a data set is transformed into a covariance matrix. This often occurs after the variance is normalized. Then eigenvalues and eigenvectors are found. Many of these eigenvectors are already orthogonal, because the covariance matrix is symmetric. Next, all of the eigenvectors are then transformed into an ON system, that is, an orthogonal and normalized system.

For a data vector v_i, the projection onto a unit vector u is $u'v_iu$, and the average value over all v_i is calculated as a sample-type average, $1/N\sum_{i=1}^{N} v_i = \bar{v}$. This average is denoted by $\bar{v}$. Consequently, the average of all the projections is $u'\,\bar{v}u$, and the length of

projection is $u'\ v^-$. To maximize the variance of projection, maximize $\sum_{i=1}^{N} 1/N(u'v_i - u'v^-)^2 = 1/Nu' \sum_{i=1}^{N} (v_i - v^-)(v_i - v^-)'u = u'Pu$. But there are constraints: $u'u = 1$. So, the Lagrangian is $L = u'Pu + \lambda(1 - u'u)$; differentiating L gives the following: $2Pu - \lambda\ 2\ u = 0$. Solving this equation gives $Pu = \lambda\ u$. This shows that u is the eigenvector, $u'Pu = \lambda$; to maximize u'Pu means to maximize the eigenvalue, λ.

PCA is most often cast in a probabilistic setting. The objective would be to find a vector v_n from an orthonormal set $\{v_1, \ldots, v_r\}$, which maximizes the variance: $C = v' \sum v$, where $v \perp \{v_1, \ldots, v_{j-1}\}$ and is the covariance matrix. It is also usually approximated by the sample matrix: $\sum = 1/n\left[\sum_{i=1}^{n} \left(v_i - 1/n \sum_{j=1}^{n} v_j\right)\left(v_i - 1/n \sum_{j=1}^{n} v_j\right)'\right]$. Note that the vectors v_i are column vectors in this expression.

After the normalized sample covariance matrix is found, then the following steps should be followed to estimate the PCA in this setting. Find all the eigenvalues of Σ. They are real and may be redundant. Find the associated eigenvectors for each distinct eigenvalue. These eigenvectors are orthogonal, and they should be normalized. For repeating eigenvalues, use Gram-Schmidt ortho-normalizing procedure on these linearly independent vectors. Denote the n normalized eigenvectors by w_1 w_2, and w_n. The matrix W is comprised of rows using the transpose of all the eigenvectors, that is, w_1' w_2', ..., w_n' in this order. Compression occurs if only m of these vectors are used as rows of the matrix W, where $m < n$. The eigenvectors should be ordered. Here, vectors associated with the largest eigenvalues should be placed ahead of eigenvectors having smaller valued eigenvalues. So the rows of W are comprised of row vectors of transposed eigenvectors and form a nonincreasing sequence relative to their eigenvalues. The matrix W is of dimension m by n.

The actual PCA is to be applied to a random vector $v = (a\ b)'$, where it is assumed that v has underlying probability distribution function F. It is such that the mean value exists and is zero. Additionally, the covariance matrix must exist, and it is assumed that it equals the sampled covariance matrix. Finally, it is postulated that the random samples v_j are drawn from F. In this case, the PCA is given by $V = W(v - v^-)$. This means that the average should be subtracted from the vectors themselves.

Example 5.12:

Say that $n = 2$, and sample values are given for $v_1 = (5\ \ 9)'$ and $v_2 = (1\ \ 3)'$. Then to normalize Σ, form the empirical average: $v^- = 1/n \sum_{j=1}^{n} x_j = 1/2\left[(5\ \ 9)' + (1\ \ 3)'\right] = (3\ \ 6)'$. Then the covariance matrix is approximated by the sample matrix: $\sum = 1/n\left[\sum_{i=1}^{n} \left(v_i - 1/n \sum_{j=1}^{n} v_j\right)\left(v_j - 1/n \sum_{j=1}^{n} v_j\right)'\right] = 1/2\{[(5\ \ 9)' - (3\ \ 6)'][(5\ \ 9)' - (3\ \ 6)']' + [(1\ \ 3)' - (3\ \ 6)'][(1\ \ 3)' - (3\ \ 6)']'\} = 1/2\{(2\ \ 3)'(2\ \ 3) + (-2\ \ -3)'(-2\ \ -3)\} = \begin{vmatrix} 4 & 6 \\ 6 & 9 \end{vmatrix}$

To find the eigenvalues, take det $(\sum - \lambda I)$ and set it equal to zero, thus giving the characteristic equation: $\lambda^2 - 13\lambda = 0$. The eigenvalues are zero and 13. The eigenvector corresponding to the largest eigenvalue is $w_1 = (a\ b)'$, where $(\sum - 13\ I)\ w_1 = 0$; doing this results in two equations with two unknowns: $-9\ a + 6\ b = 0$ and $6\ a - 4\ b = 0$, but they are dependent equations. Solving them gives $b = 3/2\ a$, but since $\|w_1\| = 1$ is desired,

that is, $a^2 + b^2 = 1$, substituting gives $a^2\ (1 + 9/4) = 1$, so $a^2 = 4/13$ and $b^2 = 9/13$. The normalized eigenvector associated with eigenvalue $\lambda_1 = 13$ is $w_1 = (2\ 3)'/13^{1/2}$. In a similar manner for the eigenvalue $\lambda_2 = 0$, the normalized eigenvector $w_2 = (3\ -2)'/13^{1/2}$. The complete matrix W is made up of these eigenvectors transposed, $13^{1/2} \cdot W =$

$$\begin{vmatrix} 2 & 3 \\ 3 & -2 \end{vmatrix}.$$

As mentioned earlier, the PCA is applied to a random vector $v = (a\ b)'$ with distribution function F. Also, say that the vectors v_1 and v_2 are sampled using F. Then, the PCA is $V = W(v - v^-) = \begin{pmatrix} a-3 & b-6 \end{pmatrix}'$. The 2 by 2 matrix $13^{1/2}$ W is given below followed by the 2 by 1 column vector $(v - v^-)/\ 13^{1/2.}$

$$\begin{vmatrix} 2 & 3 \\ 3 & -2 \end{vmatrix} \quad \begin{vmatrix} (a-3)/13^{1/2} \\ (b-6)/13^{1/2} \end{vmatrix}.$$

So, $V = 1/13^{1/2}\ (2a + 3b - 24 \quad 3a - 2b + 3)'$.

If compression is desired only the first column of matrix W would be kept, and then only the first tuple of V would exist. It would be a scalar in this case.#

The t-Distributed Stochastic Neighbor Embedding, t-SNE method is like PCA for dimension reduction, It is better for classification. For instance, it distorts or exaggerates clustering boundaries. It uses nonlinear operations to perform the data reduction.

5.7 Kernel principal component analysis

Direct PCA methods are not useful for random data or for nonlinear manifold type data. This is because PCA algorithms cannot distinguish between these two types of data; it treats them equally bad. However, PCA can be conducted using kernels. This is abbreviated KPCA, and it uses Gram matrices. KPCA is very useful when the data appears to lie on or hug a manifold. For the linear kernel, the method is almost identical to the usual PCA algorithm, particularly when the manifold in this case is linear. Importantly, when using a kernel, an additional nonlinear structure can be discovered. Reasons not to use KPCA are that, in some situations, overfitting might occur when using KPCA; additionally, transparency is often lost. In this case, the method becomes nonparametric. Moreover, reconstruction is not obvious, because replacing nonprincipal components is almost impossible when using KPCA. Finally, hyperparameters and actual kernels needed in KPCA are problematic. KPCA usually requires several runs in order to find useful parameters and kernel settings (Scholkopf et al., 1999).

KPCA techniques employ feature mappings to transform into nonlinear data. Although the feature maps, Φ, must be known, they are not employed directly. As usual, results are obtained by only involving the original inner product along with the kernel, K. Kernel PCA first introduces nonlinearity by producing a mapping into a higher dimensional feature space. An overview of the basic idea behind the underlying workings of the KPCA is outlined next. In short, it is to transform the original data using the kernel trick and then finding the eigenvalues and eigenvectors involving the kernel K. This is followed by

mapping these eigenvalues and eigenvectors into eigenvalues and eigenvectors of the desired covariance matrix C. Motivation behind the procedure comes from the similarity in the formulas for K and C. The similarity is outlined later.

The kernel matrix in essence is given by $K = XX'$. For instance, assume that K is an n by n matrix, therefore having n eigenvalues. Entries in K are scalar products involving the actual data, since X is a vector. Throughout, it is essential that all data is normalized, that is, it has mean zero, and eigenvectors have norm one. The covariance matrix of the normalized data is given by $C = X'X$, and it is a d by d matrix having d eigenvalues. The usual PCA method finds eigenvalues and eigenvectors directly from the matrix C. What is needed is some relationship between eigenvectors and eigenvalues of K and those from C. To obtain some insight into this relationship, assume that a is an eigenvector of K, $Ka = \lambda a$, that is, $XX'a = \lambda a$. Then multiplying on the left by X' gives $X'XX'a = \lambda X'a$, $Cv = \lambda v$, where $v = X'a$. This shows that if v is not zero, then v is an eigenvector of the covariance matrix, C. The opposite relationship also holds.

The critical identity needed in the derivation for showing that K can replace the correlation matrix C is given next. It shows the eigenvector and eigenvalue relationship. The principal step in this derivation is to obtain the formula $K^2\ a = \lambda\ K\ a$. Once this formula is obtained, divide by K, that is, multiply by the inverse of K. This can be done assuming that K is positive definite. In general, both C and K are only positive semidefinite. On the other hand, writing the identity above as $K\ (K\ a - \lambda\ a) = 0$ shows what was desired, namely that a_k is an eigenvector of K with an eigenvalue λ, because K is not zero. If compression is desired, then a deletion of eigenvectors corresponding to the smallest value of eigenvalues can be performed (Thompson, 2014).

By projecting into a higher dimensional space, nonlinear separable data can become linear.

Example 5.13:

This is an instance of how PCA benefits from applying the kernel and the kernel trick (Hornegger, 2021). Assume that feature vectors $x_1, x_2, \ldots x_n$ are in R^d. Each vector corresponds to an image having 1024^2 pixels. If PCA was performed on the feature vectors directly, then processing would be performed in the order of 2^{20} steps. However, using the KPCA, the total number of computations needed is only in the order of ^{50x50} steps.#

5.8 Singular value decomposition

The singular value decomposition (SVD) has a long history; it is best described in Stewart (1992). The SVD is a cornerstone procedure in much of numerical linear algebra and is important in describing the geometry of Euclidean space. The SVD procedure is outlined below with an example.

The SVD of a m by n matrix A is found by first expressing this matrix in the form $A = U\ D\ V'$, where the prime is the transpose operation. The SCALAR field is R in this situation. Here, U is an m by m orthogonal matrix whose columns consist of the orthonormal eigenvectors of AA'. Since the latter matrix is self-adjoint, its eigenvectors corresponding

to distinct eigenvalues are orthogonal, so they only need to be normalized. The columns of U are ordered from left to right using normalized eigenvectors. The ordering of the eigenvectors corresponds to the magnitude of the eigenvalues. Eigenvectors whose eigenvalues are larger are to the left of those eigenvectors with smaller eigenvalues. When two or more eigenvalues are equal, it does not matter which eigenvector occupies which column. However, these eigenvectors need not be orthogonal, but they are always linearly independent. Consequently, the Gram-Schmidt orthonormalization procedure will need to be applied before entering the vectors as columns in U.

In a similar manner, the n by n orthogonal matrix V should be found. In this case, the columns of V are made up of normalized, ordered eigenvectors of A′A. The ordering is exactly the same as for U. However, after the orthogonal matrix V is found, the transpose must be taken. Finally, the sparse diagonal matrix D consists of the square root of the eigenvalues on the main diagonal. These are organized in descending order corresponding to the placement of the eigenvectors in the columns of U or V. It does not matter which orthogonal matrix is used. This is because the eigenvectors in both matrices occupy column positions with the same nonzero eigenvalues. This can be seen next. Say that $A'Av = \lambda v$, for λ and v nonzero. Then, $AA'Av = \lambda Av$. As such, AA′ also has an eigenvalue λ, this time with eigenvector Av. Going the other way, if $AA'v = \lambda v$, then $A'AA'v = \lambda A'v$, so an eigenvalue for AA′ is also an eigenvalue for A′A.

Example 5.14:

Consider the 2 by 3 matrix A, which is given below along with AA′ to the right:

$$\begin{matrix} |3 & 1 & 1| \\ |-1 & 3 & 1| \end{matrix} \quad \begin{matrix} |11 & 1| \\ |1 & 11| \end{matrix}$$

As usual, the characteristic equation when solved will render the eigenvalues. Taking det $(AA' - \lambda I)$ and setting this quantity equal to zero gives the following characteristic equation: $\lambda^2 - 22\lambda + 120 = 0$. Therefore, the eigenvalues are $\lambda_1 = 12$ and $\lambda_2 = 10$. The corresponding eigenvectors are $v_1 = (a\ b)'$ and $v_2 = (c\ d)'$. For the larger eigenvalue, $[AA' - 12 \cdot I]\ v_1 = 0$; this shows that $a = b$. Using the fact that all eigenvectors must be normalized $a^2 + b^2 = 1$ shows that $v_1 = \left(1/2^{1/2} 1/2^{1/2}\right)'$. In a similar fashion, the second eigenvector is $v_2 = \left(1/2^{1/2} - 1/2^{1/2}\right)'$. Observe that these eigenvectors are orthogonal. Since the eigenvalue λ_1 is the largest, the matrix for U is found using as columns v_1 followed by v_2. So, $2^{1/2}\ U =$

$$\begin{matrix} |1 & 1| \\ |1 & -1| \end{matrix}.$$

In order to find V, the eigenvalues and the eigenvectors must be determined for A′A, which is:

$$\begin{matrix} |10 & 0 & 2| \\ |0 & 10 & 4| \\ |2 & 4 & 2| \end{matrix}$$

Like before, the characteristic equation, $\det(A'A - \lambda I) = 0$, has as solutions the eigenvalues. To find this equation, consider the determinant:

$$\begin{vmatrix} 10-\lambda & 0 & 2 \\ 0 & 10-\lambda & 4 \\ 2 & 4 & 2-\lambda \end{vmatrix} \begin{matrix} | \\ | \\ | \end{matrix} \begin{matrix} 10-\lambda & 0 \\ 0 & 10-\lambda \\ 2 & 4 \end{matrix}$$

Multiply elements on minus 45° on all three angles and add them together to obtain sum1. Next, multiply elements on plus 45° and then add them together to obtain sum2. Finally subtract sum2 from sum1, and this is the determinant. Evaluating the determinant $(10 - \lambda)(10 - \lambda)(2 - \lambda) - 4(10 - \lambda) - 16(10 - \lambda) = 0$, Accordingly $(10 - \lambda)[(10 - \lambda)(2 - \lambda) - 20] = 0 = (10 - \lambda)[\lambda^2 - 12\lambda] = 0$. It follows that the eigenvalues are $\lambda_1 = 12$, $\lambda_2 = 10$, and finally, $\lambda_3 = 0$. The corresponding eigenvectors are found as before by taking $(A'A - \lambda I)v) = 0$. For $v_1 = (a\ b\ c)'$, set the following equal to zero:

$$\begin{vmatrix} 10-\lambda_1 & 0 & 2 \\ 0 & 10-\lambda_1 & 4 \\ 2 & 4 & 2-\lambda_1 \end{vmatrix} \cdot \begin{vmatrix} a \\ b \\ c \end{vmatrix} = 0$$

This gives $-2a + 2c = 0$, $-2b + 4c = 0$, and $2a. + 4b + 2c = 0$; the solution is $a = 1$, $b = 2$, and $c = 1$. Next, normalizing gives the eigenvector $v_1 = 1/6^{1/2}\ (1\ 2\ 1)'$. In a similar manner, using $\lambda_2 = 10$ in the aforementioned matrix instead of λ_1 results in the corresponding normalized eigenvector $v_2 = 1/5^{1/2}\ (2\ -1\ 0)'$. Lastly, using the third eigenvalue leads to the final eigenvector $v_3 = 1/30^{1/2}\ (1\ 2\ -5)'$. Notice that all these eigenvectors are mutually orthogonal. The three eigenvectors now become the columns 1, 2, and 3, in the order given to form the orthogonal matrix V. The transpose of this matrix is needed; therefore $V' =$

$$\begin{vmatrix} 1/6^{1/2} & 2/6^{1/2} & 1/6^{1/2} \\ 2/5^{1/2} & -1/5^{1/2} & 0 \\ 1/30^{1/2} & 2/30^{1/2} & -5/30^{1/2} \end{vmatrix}$$

The only matrix left to find is D. It consists of the square root of the nonzero eigenvalues on the main diagonal, that is, starting on the top left corner and along a minus forty-five degree line. The square roots of these eigenvalues from largest to smallest populate this line. The only other thing is that $A = U D V'$; the dimensions must agree. Since the matrix dimensions are for A, two by three; for U, two by two; for D, two by two; and finally for V′, three by three, accordingly a row or column of zeros is needed to make the dimensions hold. Thus, in this example, an extra column of zeros is needed in the SVD, so D =

$$\begin{vmatrix} 12^{1/2} & 0 & 0 \\ 0. & 10^{1/2} & 0 \end{vmatrix}.$$

This is the singular value decomposition for A. The columns of U are the left singular vectors, and the columns of V are called the right singular vectors.#

A final word on the SVD is that it provides an interesting geometric interpretation when operating on a vector x. Let $A = UDV'$, an m by n vector. A major assumption will be that U and V are in the special orthogonal Lie group SO(m, R) and SO(n, R), respectively. Therefore, they represent rotation matrices; their determinants are equal to one. In

this case, first when U is applied to x, a rotation of x occurs. Let U(x) = y. Next, when D operates on y, if D is a scalar matrix, then D enlarges or shrinks y and changes the dimension. Otherwise, it enlarges certain tuples within y, compresses other tuples within y, and changes the dimension. Use D(y) = z. Finally, V* applies a rotation to z, yielding w. Accordingly, V*(z) = w. A most interesting case is when n = m = 2. Illustrations of the procedure can graphically be provided in this situation.

References

Hollings, C., Lawson, M., 2017. Wagner's Theory of Generalized Heaps. Springer books.
Horn, R., Johnson, C., 1991. 978-0521-46713-1 Topics in Matrix Anylysis. Cambridge U. Press.
Hornegger, H., 2021. Pattern Recognition Lab, Lecture Pattern Recognition.
Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. J. Educa. Psychol. 24 (6), 417–441.
Loeve, M., 1977. GTM Probability, vol. 45. Springer.
Marsden, 2003. Differential Forms and STOKES Theorem. Caltech.
Pearson, 1901. On lines and planes of closes fit to systems of points in space. Philosophical Magazine.
Scholkopf, et al., 1999. Kernel principal component analysis. Advances in Kernel Methods-support Vector Learning. MIT Press.
Stewart, G., 1992. On the early history of the singular value decomposition, IMA o reorient # 952.
Thompson, D., 2014. Nonlinear Dimensionality Reduction KPCA, JPL-Caltech Virtual Summer School Big Data Analytics.
Yano, K., 1957. The theory of Lie derivatives and its applications, North = -Holland, 978-0-7204-2104-0.
Zubelevich, O., 2012. Elementary Proof of the Cartan Magic Formula. University of Texas.

This page intentionally left blank

CHAPTER

6

Quantum annealing and adiabatic quantum computing

6.1 Schrödinger's characterization of quantum

The Schrödinger view of quantum is that of wave mechanics. His vision of quantum is most useful in fabricating qubits, sensing qubits, controlling qubits, as well as transporting them. Because of this, a little analysis is required beginning with the Schrödinger equation. This equation is a linear partial differential equation representing the dynamics and kinematics of the wave function in an isolated quantum system. In its simplest form, it involves a single nonrelativistic particle in one dimension. The equation in this case is $ih\ \partial\Psi(x,\ t)/\partial t = -h^2/(2\ m)\ \partial^2\Psi(x,\ t)/\partial x\partial x + V(x,t)\ \Psi(x,t)$. The wave function Ψ: R x R$\rightarrow$C, where C is the field of complex numbers. In the front of the wave equation mentioned earlier, the square root of minus one is multiplied by h. Here, h is the reduced Planck's constant; it is to the left of the partial derivative of Ψ with respect to time t above. The particle mass is m, and V(x, t) is the potential in the isolated system. In the beginning of this chapter, mainly the kinematics is of most importance. As such, the time-independent Schrödinger equation will be first employed. It is $h^2/2\ m\ \partial^2\Psi(x)/\partial x\partial x = V(x)\ \Psi(x)$. In a later section of this chapter, specifically Section 6.11, the adiabatic theorem is described with time time-varying wave functions.

Additionally, throughout this chapter, the state of a quantum system is assumed to be the qubit, or ket, $|\Psi>$. It is an element from a separable Hilbert space, H. Notation-wise, depending on the context, $|\Psi>$ and Ψ are used interchangeably. Moreover, this vector is always normalized, that is, $\|\Psi\|^2 = <\Psi, \Psi> = 1$. The equation, $h^2/2\ m\ \partial^2\Psi(x)/\partial x\partial x = V(x)\ \Psi(x)$, is often written as an eigenfunction equation, that is, $H\ |\Psi> = E\ |\Psi>$. In this case, E is energy of the system, and H is the Hamiltonian operator. The Hamiltonian converts state information into energy information. Accordingly, for this equation, E is understood as an eigenvalue and $|\Psi>$ is the eigenvector; it is never equal to zero. The eigenvector Ψ is frequently called the stationary state. An example of the Hamiltonian and a solution involving a very special case of the time-independent Schrödinger equation is given later.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00021-7

Example 6.1:

The harmonic oscillator describes the eigenfunction equation for photons; it involves the Hamiltonian H and the energy eigenvalues E_n, n = 0, 1, 2, …. The equation is as follows:

$H\,\Psi_n(x) = 1/2\,(-h^2/m\,d^2/dxdx + m(w_0\,x)^2)\,\Psi_n(x) = E_n\Psi_n(x)$, where $\Psi_n(x)$ is the eigenfunction and w_0 is the frequency of the oscillator. The objective is to solve this equation for Ψ_n, involving all nonnegative integers n. These solutions will be derived in a later chapter in this document. The solutions Ψ_n involve the Hermite polynomials. Hermite polynomials are described in Appendix A7, in terms of the Sturm-Liouville differential equation. Most importantly, solutions only exist for discrete energy levels, E_n, which are equally spaced. Specifically, $E_n = hw_0\,(2\,n + 1)/2$, n = 0, 1, 2, …. These solutions are found by making substitutions and using the ladder operators along with commutation relations. The operators involve an algebra using the creation and annihilation functions. However, these algebraic details will be described in the MSA framework, also in a later chapter.

Only the solution for the simplest and the most important state, the ground state Ψ_0, will be illustrated. That is, the solution for n = 0 is detailed here. The differential equation for the ground state is $x\,\Psi_0 + h/(m\,w_0)\,d\Psi_0/dx = 0$. This is a linear first-order differential equation, and it is separable. Therefore, cross-multiplying and writing, $d\Psi_0/\Psi_0 = [-x\,(m\,w_0)/h]\,dx$. Then integrating both sides of this equation gives $\ln(\Psi_0) = -\,m\,w_0\,x^2/(2\,h) + c$. Consequently, $\Psi_0 = A\,e^{-mw_0x^2/(2h)}$. The constant A will be found since, as mentioned earlier, all states must be normalized. Therefore, $\|\Psi_0\|^2$ has to equal one.

Not previously mentioned is that the Hilbert space employed in most analysis settings is a subspace of the Lebesgue square-integrable functions, L^2. These spaces appear as carrier sets in the lower view of the MSA, and are described in subsequent chapters. In short, however, since the ground state solution is trivial, the computation to normalize Ψ_0 is easily obtained: $\|\Psi_0\|^2 = <\Psi_0, \Psi_0> = \int \Psi_0^*\Psi_0 dx = \int \Psi_0\Psi_0 dx = A^2 \int e^{-mw_0x^2/(h)}dx$. Notice that this is a Gaussian kernel. As such, it can be normalized, and thus, the area of a Gaussian kernel must be set to one. The variance for this kernel involves m, h, and w_0. After normalizing, then by substitution, the value A is easily found. Thus, the final result is that the ground state solution is $\Psi_0 = (m\,w_0/(\pi\,h))^{1/4}\,e^{-mw_0x^2/(2h)}$.#

Example 6.2:

This example will reinforce the fact that state vectors must have norm one. Again the Hilbert space is L^2, and the carrier set for SCALAR is the complex field. Let the wave function be $\Psi(x) = a(x + 1)\,\chi_{[-1,1]}(x)$, where a is in C. Accordingly, the wave function $\Psi(x)$ is of compact support in the interval [-1, 1]. The adjoint $\Psi^*(x) = a^*\,(x+1)\,\chi_{[-1,1]}(x)$, to find a, set $a\int_{-1}^{1} |a|^2(x+1)^2dx = |a|^2\left(x^3/3+x^2+x\right)|_{-1}^{1} = |a|^2 8/3$. Setting this quantity equal to one gives $|a| = \left(3/8\right)^{1/2}$.#

6.2 Quantum basics of annealing and adiabatic quantum computing

Adiabatic quantum computing (AQC) is used in solving optimization problems. It uses a prepared state and a Hamiltonian, which is slowly evolved such that the system remains in the ground state. Most importantly, it is required that the problem to be solved is formulated as an energy minimization equation. In this case, the quantum annealer will try to minimize the objective function by a searching technique and actually minimize the energy. AQC inspects the problem space, or more precisely, it traverses the energy landscape searching for an energy minima. The process often avoids being stuck at a local minima by moving to other parts of the problem space. Actions occur by conventional methods as well as employing the tunneling process. When qubits tunnel together, the process is called correlation length tunneling. Changes in energy cause movement in the problem space, and quantum fluctuation refers to the process of reaching the lowest energy state. This often corresponds to the optimal desired solution. A slightly more descriptive presentation of AQC follows.

As mentioned earlier, starting in an easy-to-prepare ground state with Hamiltonian H_0, this state is adiabatically evolved. Accordingly, it is very slowly transformed into a new desired optimal state with Hamiltonian H_1. Mathematically, this might be represented as a convex combination of the two Hamiltonians. For values of s in the interval [0, 1], then the convex combination results in $H = s\ H_0 + (1 - s)\ H_1$. As the parameter s slowly and monotonically increases, from 0 to 1, H_0 goes to the desired optimal Hamiltonian H_1. Actually, the Hamiltonian H_1 corresponds to the lowest energy solution. Again it is the ground state. This is a result of the quantum adiabatic postulate. AQC exploits the quantum environment. Specifically, AQC takes advantage of quantum interference, quantum superposition, multiple qubit entanglement, and tunneling. The first three quantum effects are described next, along with examples. An account of quantum tunneling is provided in the next section, Section 6.3.

Quantum interference has to do with the possibility that wave functions of particles can reinforce or annihilate each other. Interference also has to do with the superposition of qubits and the premature collapse of this effect. Because the particles are often represented as waves, like sine and cosine waves, interference can be understood as waves overlapping each other. Waves have different frequencies; they add and subtract possibly at random. They can reinforce or cancel desired effects. Additionally, noise from the environment very often can cause quantum interference. An example of quantum interference is given below.

Example 6.3:

Quantum interference will be illustrated for a single qubit in C^2. Begin with a simple state, for instance, $|0>$. After exposing the qubit to the environment, the probability of starting with $|0>$ and winding up again in the state $|0>$ will be found. A well-known method to model the environment is to employ operators to the qubit. Here, this state undergoes various unitary transformations to mimic the effects of interference. For instance, a well-known sequence of operations is performed to convert $|0>$ into a linear combination of $|0>$ and $|1>$.

The state $|0>$, in this application, is often called the ground state. Also, $|1>$ in this case is referred to as the excited state. The sequence of operators is the Hadamard transform, H, the T or phi transform, and then the Hadamard transform one last time. When applied in that order, the sequence of operators maps the qubit, in state $|0>$, into the state, $|q>$. Chapters 16 and 17 are dedicated to describing the theoretical and applied operations in a quantum gate computer. These include the H and T operators. In any case, after applying the H-T-H transformation, the result is $|q> = 1/2\ [\|0> + |1> + e^{i\phi}\ (|0> - |1>)] = \cos(\phi/2)\ |0> - i\sin(\phi/2)\ |1>$.

Of most importance in the aforementioned analysis is that the environment being modeled caused superposition to occur. The probability amplitude for $|0>$, which was initially equal to one, is transformed and now becomes $\cos(\varphi/2)$. The probability density of obtaining $|0>$ is $P(|0>) = \cos^2(\varphi/2)$. Therefore, the probability of $|0>$ now can vary anywhere from zero to one, as given by the cosine squared function. As mentioned previously, this is a simulated effect of the environment possibly causing interference and making the probability of observation variable.#

A single quantum qubit, in C^2, $|q> = a\ |0> + b\ |1>$, $|a|^2 + |b|^2 = 1$, is in both states $|0>$ and $|1>$ at the same time. This is called quantum superposition. An illustration of superposition was provided in the previous example. There, it was seen that a single qubit in C^2, $|q> = \cos(\phi/2)\ |0> - i\sin(\phi/2)\ |1>$. Accordingly, this qubit is in both states at the same time, with proportions provided by the squares of the sinusoids.

Superposition of qubits occurs when there is more than one qubit. Two bits in the quantum world correspond to two simple qubits, $|p>$ and $|q>$, in a tensor representation in C^4. That is, $|w> = |p> \otimes |q> = |p\ q>$. For instance, in C^4, using the zero one basis, $|w> = a\ |0\ 0> + b\ |0\ 1> + c\ |1\ 0> + d\ |1\ 1>$. Similar to before, $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. Superposition now means that $|w>$ is in all four states simultaneously. Likewise, a three-tensored qubit in C^8 is in each of the eight states at the same time. In all these situations, Born's rule applies. That is, when any of these qubits are observed, only a single state appears along with an associated probability. This artifact is attributed to the wave nature of the particle. Due to the wave interference, there is an increase in the chances of obtaining a single desired state. Simultaneously, wave interference decreases the chances of obtaining other states.

Entanglement is an attribute associated with two qubits or more than two qubits. It always implies a very high correlation among the qubits. By observing a single entangled qubit, the other qubits' attributes are also known. From a slightly more rigorous point of view, assume that there are two qubits, $|v>$ and $|w>$ each in C^2. The qubit $|q>$, made up of $|v>$, and $|w>$ are said to be entangled if in $C^4 = C^2 \otimes C^2$, $|q>$; however, the qubit itself cannot be written as the simple tensor product, $|v> \otimes |w>$. In a subsequent chapter, a deeper interpretation is provided in terms of the density operator.

Example 6.4:

An example of two vectors in C^2, which are entangled in C^4, involves $|0>$ and $|1>$. In C^4, consider $|q> = 1/2\ (|0> \otimes |1>) + 3^{1/2}/2\ (|1> \otimes |0>) = 1/2\ (|0\ 1> + 3^{1/2}/2\ |1\ 0>)$. As mentioned earlier, a subsequent chapter will illustrate that an entangled tensor has the property that the product a11 a22 does not equal the product a12 a21, where $|q> = a11\ |0\ 0> + a12\ |0\ 1> + a21\ |1\ 0> + a22\ |1\ 1> = 1/2\ (|0\ 1> + 3^{1/2}/2\ |1\ 0>)$.

This criteria is described in Section 7.4. In the present situation, for $|q>$, a11 = 0, a12 = 1/2, a21 = $3^{1/2}/2$, and finally a22 = 0. So a11 a22 = 0, but a12 a21 = $3^{1/2}/4$, and so there exists entanglement.#

Adiabatic quantum computing can perform quantum annealing (QA), as well as provide an alternative method for quantum computing utilizing unitary gates. AQC was shown to be equivalent to the standard unitary gate model of quantum computing (Aharonov et al., 2014). However, the methods are very different. The objective in AQC methods is to find a global minimum (Krauss, 2013), under adiabatic conditions. Beginning with initial conditions for the state $|\psi>$, a continuous time evolution of the Schrödinger equation is employed in finding the final observed value. This is what AQC is all about. By not controlling the environment properly, the final state may only be a local minimum. As usual, the Schrödinger equation is ih $\partial|\psi>/\partial t = H(t)\ |\psi>$. It is evolved under adiabatic conditions for the Hamiltonian H, for t in [0, T], where T is the total evolution time.

As previously mentioned, the adiabatic condition guarantees invariant eigenstates for the Hamiltonian. Here, it is assumed that there is slow time evolution. Additionally, it is assumed that there is not too small of a gap of energy between the populated eigenstates and all other excited energy states. This gap governs the computational complexity of AQC. Smaller gaps induce longer computation times. AQC is known to provide a quadratic speed up in finding an optimum solution, over conventional algorithms. Moreover, it keeps qubit coherency and provides noise immunity better than other quantum computation models (Childs et al., 2001a,b).

Basically, there are two types of methods for controlling the annealing process: quasistatic control and coherent control. Quasistatic control has an annealing time larger than thermal relaxation time, resulting in a thermal equilibrium for most of the evolution. In coherent control, the annealing time is less than relaxation or de-coherence time; in this case, Schrödinger equation prevails.

Superconducting electronics is one of the methods for AQC and QA implementation (Wendin, 2017); in this case, the Josephson junction is utilized. It employs tunneling effects. Trapped ion technology is also used for AQC (Zhang et al., 2018).

These inputs are converted into electrical currents, electrical voltages, and magnetic fields. The electromagnetic sources control the qubits, which always begin in a superposition type state. As the annealing process proceeds, the qubit spins evolve trying to determine the lowest energy state. This process utilizes quantum interference, superposition, and entanglement as outlined earlier. Additionally, these qubits employ the tunneling methodology described in Section 6.3.

For an arbitrary particle, qubit, or wave function Ψ, the probability amplitude is the value, Ψ. Additionally, when Ψ is observed, the Born rule is utilized in defining the actual probability density; it is P (ψ) = $\|\Psi\|^2$.

6.3 Delta function potential well and tunneling

The main content of this section is the tunneling effect. In short, a particle can take a shortcut and burrow through barriers instead of climbing over a barrier. Thickness does not matter; a particle can pass through a substance faster than light can travel through a

vacuum. Two situations exist in this section. If $E < V$ states, bound states will arise for a delta function potential, and when the inequality is reversed, tunneling occurs. In the latter case, free particles are said to exist. They are not normalizable since they are not square integrable in general. Moreover, the spectrum is usually continuous since the states are not bound. This section is based on the excellent source (Carson, 2014).

Bound states go to zero as $|x|$ goes to infinity, and they are normalizable. Additionally, they have energy levels that are usually quantized and discrete. This is a result of boundary conditions using the time-independent Schrödinger equation. In a sense, the concept of bound state is the opposite of a scattering state in which tunneling occurs. In a bound state, the energy of a particle in that state is less than that of the potential energy at infinity. Moreover, bound state functions are standing waves. The spectrum of a bounded state is always discrete. Moreover, the corresponding eigenfunction decreases exponentially for large values of $|x|$. These concepts are described later involving examples of bound states as well as scattering of states.

Solutions of time-independent Schrödinger equations must match up at boundaries for piecewise solutions. In general, the wave function ψ must be continuous. When there is no delta function, the derivative $d\psi/dx$ must also be continuous. However, if at some discrete point x_0 the derivative does not exist, then if $V(x_0) \to \infty$, integrating will lead to a solution (Zwiebach, 2017).

The infinite potential well is illustrated in Fig. 6.1A and B. At the origin, there exists a delta function of value $-a$., $V(x) = -a\delta(x)$. There are two cases, depending on whether the energy E is negative or positive. In the former case, illustrated in Fig. 6.1A and described in the next example is the bound state situation. Correspondingly, the energy E is less than zero. The second example illustrates the free particle solution and demonstrates the tunneling process.

Example 6.5:

The bound state solution is found by solving the time-independent Schrödinger equation: $-h^2/(2\,m)d^2\psi/dx^2 = E\,\psi$. Here, $V = 0$ to the left and right of the origin. Also $E < 0$. Make a substitution $d^2\psi/dx^2 = k^2\,\psi$, where $k = (-2mE)^{1/2}/h$, and do not forget that $E < 0$.

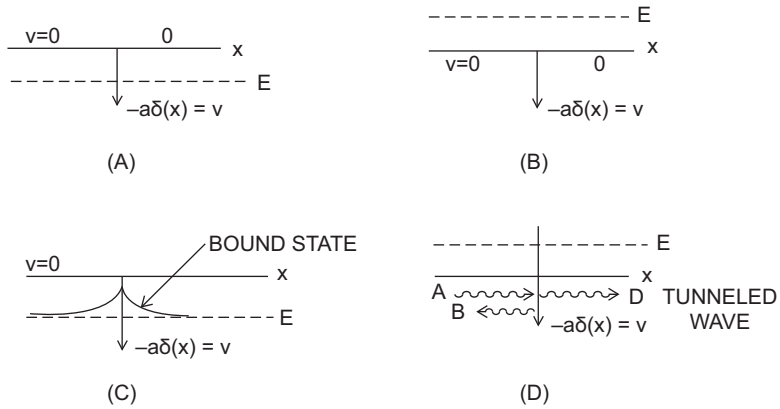


FIGURE 6.1 Bound state and scattering state tunneling effect. (A) Bound state, (B) Scattering state, (C) Normalized bound state solution, (D) Tunneling effect.

In this case, two negatives make a positive. Solving in general, $\psi(x) = A\ e^{-kx} + B\ e^{kx}$. The delta function exists at the origin; to the left of the origin, the solution is ψ_1; and to the right, the solution is ψ_2. The delta function at the origin is $-a\delta(x)$. The quantity B must be zero for $x>0$, since it provides an unbounded solution. For $x<0$, the scalar A must be zero.

Since the waveform must be continuous, boundary matching is applied. So, for continuity $\psi_1(x) = B\ e^{kx}$ must be equal to $\psi_2.(x) = A\ e^{-kx}$ for $x = 0$; this implies that the two constants A and B are equal, that is, $A = B$.

Derivative matching fails because of the delta function infinite potential. Instead, integrate the time-independent Schrödinger equation from minus epsilon to epsilon. And let epsilon go to zero. Using $V(x) = -a\ \delta(x)$, then $\int_{-\epsilon}^{\epsilon} -h^2/(2m)\ d^2\psi/dx^2 dx - \int_{-\epsilon}^{\epsilon} a\delta(x)\psi(x)\ dx = \int_{-\epsilon}^{\epsilon} E\psi(x)dx$. Integrating and preparing to substitute limits of integration gives $-h^2/(2m)d\Psi/dx|_{-\varepsilon}^{\varepsilon} - a\psi(0) + E\psi(x)|_{-\varepsilon}^{\varepsilon} = -h^2/(2m)d\Psi/dx|_{-\varepsilon}^{\varepsilon} - a\Psi(0) + 0$. The last zero occurs from the integral mean value theorem. So $d\psi/dx|_{-\epsilon}^{\epsilon} = -2ma/h^2\ \psi(0)$. Considering the left and the right solutions and differentiating gives $d\psi_1(x)/dx\ |_{-\epsilon} = B\ k\ e^{-k\epsilon}$ and $d\psi_2.(x)/dx|^{\epsilon} = -Bk\ e^{-k\epsilon}$. Subtracting the lower limit from the upper limit and letting $\epsilon \to 0$ yields $-2B\ k = -2\ ma/h^2\ \psi(0)$. Let $\psi(0) = B$ and then cancel out B, so $k = ma/h^2$, but $k^2 = -2\ mE/h$, and this shows quantization and allows solving for the quantized energy $E = -ma^2/(2h^2)$.

Finally, normalize the wave function $\int_{-\infty}^{\infty} \psi^*\psi\ dx = 1$, using the fact that this is a Gaussian-type kernel and the area under the curve is one. This determines $B = (ma)^{1/2}$. The normalized bound state solution is described by $\psi(x) = (ma)^{1/2} e^{-ma|x|/h^{1/2}}, E = -ma^2/(2h^2)$. This solution is illustrated in Fig. 6.1C.#

The next example below illustrates the free particle solution and demonstrates the tunneling process. In this situation, the energy E is greater than zero.

Example 6.6:

Scatter the free particle wave solution for the case $E > 0$. The value V is zero everywhere except at the origin. The time-independent Schrödinger equation is $-h^2/(2\ m)d^2\psi/dx^2 = E\psi$. This time, since the energy is positive, let $k^2 = 2\ mE/h^2$; then $d^2\psi/dx^2 = -k^2\psi$. Again, there are two separate solutions one to the left of the delta function and one to the right. To the left of the origin, the solution is $\psi_1(x) = A\ e^{ikx} + B\ e^{-ikx}$, and to the right of the origin, the solution is $\psi_2(x) = D\ e^{ikx} + F\ e^{-ikx}$. Using continuity of the wave function solutions amounts to setting the left-hand solution equal to the right-hand solutions at the origin. This shows that $A + B = D + F$.

As mentioned in the previous example, the derivative need not be continuous at the origin since there exists a delta function potential. So as before, integrate the Schrödinger equation. As in the case of a bound state, the limits form a small neighborhood about the origin. Thus, the integral is $\int_{-\epsilon}^{\epsilon} -h^2/(2m)d^2\psi/dx^2 dx - \int_{-\epsilon}^{\epsilon} a\delta(x)\ \psi dx = \int_{-\epsilon}^{\epsilon} E\psi dx$. Applying limits gives $-h^2/(2\ m)d\psi/dx\ |_{-\epsilon}^{\epsilon} - a\psi(0) + E\ \psi(x)\ |_{-\epsilon}^{\epsilon} = -h^2/(2m)d\psi/dx\ |_{-\epsilon}^{\epsilon} - a\psi(0) + 0$. Again, the last zero results from the mean value theorem for integrals. Differentiating and substituting in for the left- and right-hand solutions shows that $ik\ (D - F) - ik\ (A - B) = -2\ ma/h^2\ (A + B)$, where $\psi(0)$ was arbitrarily chosen to be equal to $\psi_1(0) = A + B$. Define

a constant to compress the solution, where $\beta = ma/(kh^2)$. So, $D - F = A(1 + 2i\beta) - B(1-2i\beta)$.

Now, heuristics can be applied. Say that scattering occurs only from the left. Assume that a wave ψ is transmitted from left to right. Therefore, it can be assumed that the constant A is known. So all solutions will involve A. Also, in this case, B can be thought of as being a coefficient for a reflected wave. Since the wave is going from left to right, F should be set to zero because it cannot be reflected. However, D can be considered as the coefficient for a tunneled wave. Therefore, the overall solution will be given in terms of A. Solving for B and D results in $B = I\beta/(1 - i\beta)\ A$ and $D = 1/(1 - i\beta)\ A$.

The reflection coefficients are defined using $R = |B|^2/|A|^2 = \beta^2/(1 + \beta^2)$. Similarly the transmission coefficient, that is, the tunneling coefficient, is $T = |D|^2/|A|^2\ 1/(1 + \beta^2)$. Replacing β by $ma/(kh^2)$ results in $R = 1/(1 + 2\ h^2E/(ma^2))$ and $T = 1/(1 + ma^2/(2h^2E))$. Moreover, $F + T = 1$. To illustrate quantum tunneling, as $E \to \infty$, $T \to 1$, that is, the greater the energy the more likely the particle will tunnel through. On the other hand, the closer the energy is to zero, the more likely that the particle will be reflected. In the nonquantum situation, that is, classically, these values are equal to $T = 1$ and $R = 0$ for the potential well. Also these values are $T = 0$ and $R = 1$ for the potential barrier. However, in quantum theory, for the delta function situation, they have the same value. That is, if a barrier was used in this problem instead of a quantum well, the solutions would be the same. That is, $T = 1/(1 + \beta^2)$, and $R = \beta^2/(1 + \beta^2)$. Again, these quantities sum to one when added together, and they are interpreted as being probabilities. Fig. 6.1D illustrates the original wave, the reflected wave, as well as the tunneled wave.#

6.4 Quantum memory and the no-cloning theorem

Storage of qubits is difficult due to its coherence time. This is the amount of time that it takes for quantum memory to sustain the superposition before collapsing.

In quantum computing, several qubits correspond to a large number of states. For instance, n qubits represent 2^n states. Storing these qubits is another story due to interference from the environment including temperature, as well as interference from other qubits. In short, the quantum bit has a tendency to collapse or de-cohere. The entanglement of qubits is also lost in this case. Most critically, during observation or measurement more often than not, information about the qubits disappears. A single state with an associated probability is always the outcome, as mentioned previously. The Holevo's theorem also shows that even though there might be 2^n states, only n data bits can be utilized. Finally, reproducing qubits already existing in memory is not possible, due to the no-cloning theorem detailed later.

The no-cloning theorem specifies no replication of states in a finite-dimensional Hilbert space H of rank greater than one. Specifically, there is no linear map $T: H \to H \otimes H$, such that $T\ |\psi>$ produces the multiple, $|\psi>\ |\psi>$ for all ψ in H. This result can easily be shown using contradiction. So, here it is assumed that the qubit $|\psi>$ can be replicated using T, as the linear replication operator. Consider a scalar a in $C - \{0\}$, and say that $T\ |\psi> = a^{-1}|\psi>\ |\psi>$. Use two linear independent vectors $|u>$ and $|v>$ in H, and scalars b and c in C. Consider the superposition of these two states $|\psi> = b\ |u> + c\,|v>$.

Then, by the assumed replicating property of T and linearity, the following expressions are equal, a [T (b |u> + c |v>)] = (b |u> + c |v>) (b |u> + c |v>) = b^2 |u> |u> + b c |u> |v> b c |v> |u> + c^2 |v> |v>. But, also due to the linearity of T, the following holds: a [T (b |u> + c|v>)] = a b T |u> + a c T |v> = b |u> |u> + c |v> |v>. Next, equate this expression with the previous expression. Then equating scalars gives $b^2 = b$, $c^2 = c$, and bc = 0. So at least one, c or b, equals zero. Consequently, this implies that $|\psi>$ is a multiple of |u>, or it is a multiple of |v>. In any case, the result is that $|\psi>$ is one dimensional, that is, a single dimension only exists.

Example 6.7:

If the Hilbert space is H = C, over the complex field, then linear map T: H→H @ H, such that T $|\psi>$ is a multiple of $|\psi>$ $|\psi>$ for all ψ in H. That is, the no-cloning theorem does not hold in H. So the vectors or the complex number in C can be reproduced. This follows since for b in C, b T $|\psi>$ = T b $|\psi>$, so that b $|\psi>$ $|\psi>$ = b^2 $|\psi>$ $|\psi>$, implying that, for instance, b = 1 holds true. That is, for any vector, v or actually a scalar v, in C, the operator T duplicates this value, namely T (v) = (v v)′ in C^2.#

Although the no-cloning theorem holds, numerous companies invest in creating quantum memory devices. Several recent approaches in memory development are described next.

The Qunnect Company reported on demand electromagnetically induced transparency quantum-type memory (Finke, 2021). The impact of this quantum memory will also help solve other problems besides new device development and applications in machine learning.

Single light particles, photons, are often used as qubits, particularly when employed in networks. They are utilized in the transmission of qubits mainly by fiber optics. Impurities within the fiber cause qubit loss and errors. Additionally, loss is a function of the length of the fiber. Quantum memory sometimes utilizes photons. Here, quantum bits encoded in photons are stored without observation or measurement. In this case, at temperatures of four degrees kelvin, photonic information stored on an electron was stored on a silicon nucleus (Stas, 2022).

Different approaches for creating quantum memory include dilute atomic gases and rare earth ions embedded in glass. Also silicon-vacancy centers (SiVs), which are quantum bits made up of electrons around single silicon atoms, are embedded in diamond crystals (Stas, 2022). Crystals are used in storing photonic qubits. In this case, a duration of 20 milliseconds record was achieved for solid-state quantum storage, without nonentanglement of the atoms (Ortu et al., 2022).

6.5 Basic structure of atoms and ions

Before the Bohr model of an atom is described, it should be mentioned that the location of atoms within an atom is not exact; they are described in terms of probabilities. This is the Schrödinger interpretation; the location of quantum particles forms an electron cloud. Electrons do not travel in fixed paths; they travel instead in regions called electron clouds.

The revolution about the nucleus is wave-like and are three dimensional and are called orbitals.

Energy levels inside atoms are of uneven levels. However, they are roughly described using the Bohr model. This model explains how electrons can jump orbits only by absorbing or emitting energy. Bohr had a probabilistic interpretation, but it is conditional. Namely, there are probabilities associated with particles, but only if the particles were not observed. The Bohr model indicates that the amount of energy within an orbit is proportional to its size. Electrons revolve around neutrons and protons; together they constitute the nucleus. In general, they rotate without emitting or absorbing energy. Energy shells correspond to energy levels having different orbit energy levels. Orbits are labeled K, L, M, and N. Stationary orbits consist of electrons within stable orbits. They do not radiate energy; the energy in the orbit is fixed. Only certain orbits are stable; the ones that are not produce radiation. Electrons are never stationary within an atom; otherwise, the electron would fall into the nucleus. The farther away the electron is from the nucleus is an indication of a lesser attraction force. The electron potential energy goes like $1/r$, where r is the distance to the proton.

Photons of energy are emitted when an electron drops to a lower energy level. The energy of the photon equals the difference in the energy levels before and after the jump. The Bohr model could not explain the details of the hydrogen and helium spectrum. The energy is classified as having sharp, principal, diffuse, fundamental (SPDF) electrons. This is a method of characterizing spectral lines in accordance with the Pauli exclusion principle. The Pauli exclusion principle specifies that fermions such as electrons can only exist with no more than one particle within a state. However, two particles can exist, but with distinct spins. For SPDF electrons, subshells have orbitals that occupy different energy levels. The maximum number allowed in S is two electrons, in P six electrons, D has ten electrons, and F has 14 electrons. Orbitals are labeled in order as 1, 3, 5, and 7. Multiplying by 2 gives the maximum number of electrons per shell.

Quantum numbers allow the unique identification of a particular electron. These are given in order, n, L, m_L, and m_s, the energy, angular momentum, magnetic momentum, and angular momentum, respectively. The principal quantum number is n; it identifies the distance from the nucleus to the electron as well as the energy. The orbital angular momentum quantum number is L; it identifies the shape of the orbital. The magnetic quantum number is m_L, which describes the orientation of the orbital. The electron spin quantum number is m_s; it indicates a clockwise, or a counterclockwise rotation. The corresponding spins are denoted as $1/2$ or $-1/2$, also called spin down or spin up, respectively. Previously mentioned, a maximum of two electrons can occupy the same orbital; in this case, one has spin up and the other has spin down. This is a consequence of the Pauli exclusion principle, which is a fundamental concept in Section 15.3, on Fermion Fock space.

The electronic configuration also uses subshell labels. It specifies the principal quantum number n, which identifies the electron shell. This is followed by a letter S, P, D, or F, which tells the type of orbitals within the subshell. Finally, there is a superscript indicating the number of electrons within the subshell. Electrons fill lower shells before they fill higher ones. This is called the Aufbau principle or the building-up principle (Scerri, 2013).

Atoms that gain or lose electrons are called ions. When atoms lose an electron on its outer shell, the atom becomes positively charged and is called a cation or just a positively charged ion. When an atom gains an electron, it becomes a negatively charged ion or also called an anion.

Examples of ions utilized in trapped ion qubit fabrication are given next. It must be mentioned, however, that in ion trap quantum manufacturing it is difficult to scale up the number of qubits.

Example 6.8:

The $Ca+$ ion is the calcium ion. It arises from the calcium atom by losing one or two electrons from its fourth shell, the valence shell. There exist 20 protons within the nucleus indicative of the atomic number. This ion is used in trapped ion qubit fabrication. The electronic configuration for this ion is $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6$. The exponent indicates the number of electrons per subshell. In short, it is 2, 8, 8, because $1s^2$ indicates two electrons, $2s^2\ 2p^6$ shows eight electrons, and $3s^2\ 3p^6$ again shows eight electrons. So this ion is positively charged since only 18 electrons exist within Ca^2+. If only one electron is lost in the valence orbit, then the electron configuration is $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 4s^1$, that is, 2, 8, 8, 1.#

Example 6.9:

Ytterbium, Yb, ions are also used in trapped ion qubit fabrication. There it is called Yb 171, due to its mass number, and it is an isotope with 101 neutrons. Yb has 70 protons and 103 neutrons. Yb itself has an orbit structure, 2, 3, 28, 32, 8, 2. The Yb ion loses one or two valence electrons and becomes positively charged.#

Since qubits were fabricated using trapped ions, long ago in the 1990s the main idea was to use them to make general-purpose computers. In this direction, numerous companies followed DiVincenzo's criteria on how to construct a quantum computer (DiVincenzo, 2013). These computers are usually not for specialized purposes such as QAC, which are employed mostly for optimization purposes. They are for general-purpose computers, here called gate-oriented computers. There are seven DiVincenzo's criteria, five dealing with computers and two having to do with transmitting information and the need for a quantum channel. The criteria were often called the gold standard for constructing a quantum computer. The first of DiVincenzo's criteria is to obtain a scalable system with well-defined qubits. The next benchmark is to make sure the device can initialize into a beginning fiducial state, for instance, $|0000000>$. Of paramount importance is to have a long de-coherence time for the qubits. That is the need to stay in a state long enough for completing computation. Having a universal set of quantum gates for performing computations is also a must, along with basis gates. Finally, for a computer, there has to be measurement capability. The other criteria deal with communication.

A similar set of guidelines was employed for semiconductor-type qubits for use in quantum gate computers. Early on Ga and As were employed for quantum dot qubits. Subsequently, Si and Ge took their place in quantum dot-type qubit creation.

Example 6.10:

The electron structure of some of the most prominent semiconductor materials is given below:

Silicon, Si, is a neutral atom of atomic number 14. Its electron configuration is $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^2$.

Germanium, Ge, is a chemical element and has an atomic number of 32. Its electron configuration is $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 3d^{10}\ 4s^2\ 4p^2$. In short, the electron distribution is 2, 8, 18, 4.

Gallium, Ga, has an atomic number of 31. The electron configuration is given by $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 3d^{10}\ 4s^2\ 4p^1$. It can exist as an ion; in this case, it is Ga + with electron configuration $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 3d^{10}\ 4s^2$. Also it can be configured as the ion $Ga+^3$ with electron configuration $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 3d^{10}$.

Arsenic, As, has an atomic number of 33. Its electronic configuration is $1s^2\ 2s^2\ 2p^6\ 3s^2\ 3p^6\ 3d^{10}\ 4s^2\ 4p^3$. #

6.6 Overview of qubit fabrication

Qubits are built essentially in two distinct ways: There are nonsolid-state platforms such as trapped ions and photonic methods. The second way is by using solid-state devices. In the solid state situation, the most important method is superconductors. The next most important is the quantum dot methodology. Lastly, there are methods for exploiting defects within an atom, for instance, NV centers in diamonds.

- Trapped ions are one of the original methods of creating qubits. These qubits are used to produce general-purpose, gate-oriented quantum computers. As previously mentioned, an ion is a charged particle; usually it, is an atom that lost one or more of its electrons. The first qubit in the mid-nineties was fabricated out of a single beryllium trapped ion. Lasers control the discrete energy levels of the remaining electrons. Normally, these levels are denoted by 0 or 1. Electromagnetic fields make sure that electrons remain in order, so that they can be identified. Lasers and radio waves are utilized in toggling the 0 or 1 attribute of the qubit. De-coherence is not a serious problem for trapped ion since they are not exposed to the environment. These qubits are sheltered by being kept in a dark vacuum. The problem with trapped ion qubits is that it is hard to fabricate large amounts, as also mentioned in the previous section. However, in Section 6.7, a more informative description of this type of qubit fabrication is provided. A gate-oriented 40 qubit computer was recently reported (IonQ Staff, 2023).
- Neutral atoms are atoms for which the positive and negative charges are equal. These entities are similar to trapped ions, but use light to trap atoms and hold them in position. As usual, the qubits are created and entered into the superposition states. Using neutral atoms also allows more versatile two-dimensional configurations, by not limiting the possible connections among atoms. This results in simpler methods for creating entanglement among qubits.

- Many companies employ a superconductor as their primary method for creating, controlling, and transmitting qubits. When cooled, a superconductor converts from a normal state of having electrical resistance to a state in which just about all electrical resistance disappears. Beginning with an oscillating electrical current, and then cooled, the circuit acts like a qubit. These qubits must be kept cold; otherwise, they will de-cohere. Qubits are mapped into one of the two states, 0 and 1, corresponding to distinct energy levels. State selection is controlled by radio waves. By pulsing the radio waves, quantum superposition is created. An important attribute is that the actual circuit often exists on a small microchip.

Superconducting materials are made by sandwiching a very thin insulator in-between two metals, referred to as a Josephson junction (JJ). It acts like a nonlinear inductor shunted with a capacitor. From an electrical point of view, this is similar to the harmonic oscillator briefly discussed in Section 6.1. In this device, de-coherence is problematic due to the oxidation of metal electrodes that typically cause loss of information. Another shortcoming for superconducting qubits is the need for strict control of the zero point temperature. A more in-depth description of superconducting qubits is described in Section 6.8, where an emphasis is on Andreev reflections producing tunneling of pairs of electrons.

- Quantum dots (QDs) use a semiconducting material to fabricate qubits but are very different from JJ qubits. Quantum dot qubits are created when electrons or holes are enclosed in a static potential well within a semiconductor. Qubits have states corresponding to the spin of an electron. For instance, $|0>$ denotes spin up and $|1>$ indicates spin down. Superposition of these states $|v>$ is described as usual $|v> = a\ |0> + b\ |1>$, where a and b are complex numbers. Fabrication most recently involves a silicon, Si, substrate. A quantum well is also constructed using Si. A large advantage of these types of qubits is the compatibility with a complementary metal oxide semiconductor (CMOS). The main advantage of using this type of semiconductor is that it utilizes much less power.
- Some companies exploit defects or spaces within atoms in the material lattice structure. Defects such as these may change the electron movement within materials. These defects may be a foreign atom present or missing atoms within a substance. Defects also may attract and consequently trap elections. This process enables the control of electron spin. Spin, which is rotational momentum, is employed in encoding the electron using 1 or 0 in accordance with being spin up or spin down. A consequence of using quantum defects is the little need of low temperature control, resulting in longer coherence times. One of the most prominent defect type qubits is the spin-based qubits involving NV centers, that is, NV centers in diamond qubits. In this case, a nitrogen atom is substituted for a carbon atom. This results in a NV center. Stencils were created to implement these defect patterns. With their use, coherence times have increased. An additional benefit is a greater ease in establishing qubit entanglement.

Less expensive materials are being utilized in the NV methodology in place of diamonds. Nitrogen vacancies occur in aluminum nitride. Here, it was found that control of the electron spin is possible. Additionally, silicon carbide has defects and can be used similarly to diamonds.

- Photon-type qubits usually involve polarization $|0>$ and $|1>$. They are distinguished by the types of rotations. These are H, V, D, A, R, and L, abbreviated for horizontal, vertical, diagonal, antidiagonal, clockwise rotating, R, and lastly, L for anticlockwise rotating, respectively. Orientations are controlled by the use of electric fields vertical or horizontal. Lasers are used by sending signals to certain crystals, resulting in two qubit photons being emitted as a maximally entangled pair.
- Topological qubits and topological computing utilizing Majorana qubits are also being considered. Two Majorana particles are considered a fermion. The particle itself is classified as a Bogoliubov quasiparticle-an electron void that carries charges of a hole and an electron. The Majorana particles exist at a relatively large distance apart resulting in a long de-coherence time, thus making topological computing very promising. Majorana qubits are mentioned again in Section 6.8, in reference to the Andreev effect for electron and hole tunneling.

Different companies employ distinct methods for qubit fabrication. Intel fabricated silicon quantum dots. Here, an all optical lithography was used to transfer a pattern on a substrate involving a photosensitive material. IBM mainly employs semiconducting materials such as niobium and aluminum used on a silicon substrate. This particular qubit is called a superconducting transmon qubit.

Ion trap fabrication compares to other qubit creation methods in distinct ways. Ion traps have long life, and usually superconducting qubits have a shorter life. However, silicon superconducting are easier to fabricate, whereas ion trap has long coherence time. Ion trap gate fabrication is a long process, but it might not be subject to external noise or leakage as other methods (Blinov et al., 2004; DiVincenzo, 2013).

6.7 Trapped ions

The procedure for making qubits out of ions is described slightly in more detail in this section. First an atom is chosen, for instance, calcium, Ca (Hui, 2019). Then the atoms are heated to form a vapor. For Ca, a gas state is formed at about 900 degrees c. Valence electrons are stripped out of the substance by bombarding it with photons. Next, an radio frequency (RF) or Paul trap is used to keep the atoms at a single location. At this time, doppler cooling is utilized to reduce the kinetic energy of the ions. These ions are close to a stationary state. Microwaves, lasers, and RF fields are used to control the ions. Observation of qubit states is arrived at by employing lasers of a predetermined frequency. These ions cannot be kept in stable stationary equilibrium state using static charge. This is Earnshaw's theorem (Simon et al., 1995). Accordingly, ions are trapped between electrodes, using oscillating RF along with a direct current potential.

Trapped ion technology uses the assumption that there is one electron left in the valence shell of the calcium ion. Then in the process, to construct an optical qubit, an arbitrary choice is made. Here, the preference for the ground qubit state $|0>$ is to correspond to the $4s^1$ state, using the Bohr model. It might spin up or spin down, each with equal probability. Next, an excited state must be chosen corresponding to $|1>$. By referring to the specs on the calcium atom, the distances from $4s^1$ to the closest subshells or orbitals

are listed and identified. For instance, the distance of 397 nm from $4s^1$ locates an orbit for an electron on the 4p orbital. Other orbital distances are also listed in this specification. Taking the distances into consideration along with coherent times, a choice will be made in defining the $|1>$ qubit. Photon energy is induced or released in the electron transition process. So a laser of frequency corresponding to the optical wavelength might be used in choosing an electron, for instance, in 3D orbital. The resulting qubit is designated as $|1>$. Superposition is created using Ramon transmission. It employs two laser beams resulting in the superposition of the two qubits, a $|0> + b\ |1>$.

In general, these qubits have long coherent times. They are used in large-scale quantum computers.

Ion trap qubits exist in three distinct types; there is the Zeeman ion-trapped qubit, the optical ion-trapped qubit described earlier, and the hyperfine qubit. They are also fabricated using different types of ions. Different atoms have different attributes that not only affect the manufacturing but also determine the control, susceptibility to noise, as well as the observation ability. In any case, the Zeeman trapped ion qubit has a long lifetime for an electron spin superposition state of a trapped $Ca+$ ion. More information can be found in Ruster et al. (2016).

Optical qubits and hyperfine qubits, both, are easy to measure and are made using the electronic states of an ion. They are distinguished by the qubit energy splitting level as well as couplings. The former uses two ground-state levels. The latter uses an electronic ground state and an excited level separated by microwave frequency. An optical qubit has a shorter lifetime compared to the hyperfine qubit.

Hyperfine qubits were created using two distinct energy levels: hyperfine S and F levels of the Yb 171 ion. The use of two distinct qubit types enables quantum error prevention. This minimizes cross talk error. This error has to do with scatter photons creating errors in a qubit's information content (Mattel, 2021).

6.8 Super-conductance and the Josephson junction

The concept of Cooper pairs is explained in detail. After this, super-conductance and the JJ as well as the principal cause of tunneling are described. It is called the Andreev reflection. Cooper pairs are basic for the existence of super-conductance. These pairs also called Bardeen-Cooper-Schrieffer (BCS) pairs. They consist of two electrons or other fermions that temporarily bind together at low temperatures. Attractions between electrons in standard metals cause paired states of electrons or fermions. These paired electrons are entangled and condense at very low temperatures into the ground state. Accordingly, they flow with zero electrical resistance with no scattering within superconductors; this is the BCS theory. When the entangled particles de-cohere through interaction with the environment, entanglement is broken. Broken Cooper pairs are called Bogoliubov quasiparticles. Mentioned later is the principal cause of tunneling and is called the Andreev reflection. Tunneling of a Cooper pair can be explained as a result of a negatively charged electron entering a superconductor and a simultaneous exiting of a positively charged hole.

Cooper pairs have zero sum spin and have mass and charge twice that of a single electron. The momentum in each electron within a Cooper pair is equal and opposite in sign.

They have power energy less than the Fermi energy, and as such, thermal energy can break the bonding. Large groupings of Cooper pairs are therefore found at low temperatures. Accordingly, the pair is said to be bound in conventional super-conductance caused by electron-photon interaction. The quantum effect creates super-conductance. This bunching or pairing is the Cooper or BCS pairing, but the binding is weak. Accordingly, only at low temperatures are there numerous BCS pairs (Frolov, 2014).

The Josephson junction is described following the reference: Frolov (2014). The presentation is based on phase difference between superconducting materials sandwiching a nonsuperconducting barrier. The barrier is composed of a metal N, or a dielectric D, or even a vacuum V. This substance is sandwiched between two superconductors S_1 and S_2. The resulting superconducting device is denoted as S_1 N S_2, S_1 D S_2, or S_1 V S_2, respectively. Additionally, it is assumed that the separation between S_1 and S_2 forms a narrow junction. This is illustrated in Fig. 6.2A, as presented and derived in Frolov (2014). Here, the superconducting device to the left is labeled by the wave function $\psi_1 = \mathrm{n}\, e^{i\phi_1}$, and to the right, the wave function is designated as $\psi_2 = \mathrm{n}\, e^{i\phi_2}$. The superconducting material can be thought to be a catalyst for creating a scattering wave function with the tunneling effect, as illustrated in Fig. 6.1D of Section 6.3. In the S_1 N S_2 illustration, the wave functions decrease in the barrier region, N, but not so much so that Cooper pairs cannot tunnel through the barrier region. The phase difference is given by $\Delta = \phi_2 - \phi_1$ and might not equal zero. The difference in the phase of the superconductors will be the principal quantity governing the physical properties involving super-conductance. Throughout, the process phase coherence must be maintained. A superconducting current is established; this is called the Josephson super-conductance effect. It could be considered as a result of the tunneling of Cooper pairs.

A simple illustration of the JJ effect involving the non-AC current versus voltage V is provided in the graph of Fig. 6.2B. The illustration presents a time averaged description; in this diagram, the voltage V versus the supercurrent I. It has a voltage value of zero when the

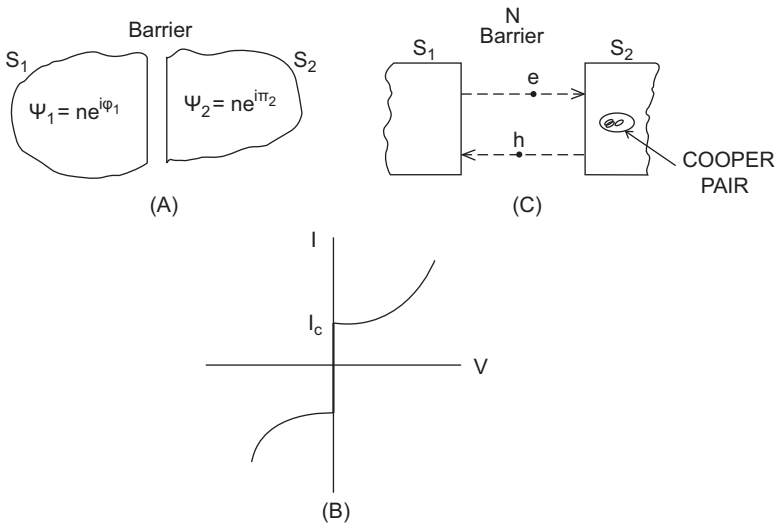


FIGURE 6.2 Cooper pair tunneling. Based on (Frolov, 2014), (A) Frolov Barrie, (B) Cooper Pair Tunneling, (C) Voltage-Current Dead Zone.

current is swept from minus the critical current $-I_c$ to plus the critical current I_c. The phase is called the supercurrent state. After the critical current is achieved, the voltage jumps and eventually becomes the normal state where it acts like a resistor. The first JJ equation involves the supercurrent I. It is created by the phase difference between the two superconductors, $I = I_c \sin(\Delta)$. When Δ changes from zero to $\pi/2$, the supercurrent phase exists.

Andreev bound state gives rise to super-conductance. Andreev reflection occurs when an electron is transmitted from the first superconductor to the second superconductor. In this case, the electron will be reflected as a hole and have energy minus its original energy. Its path will overlap its original path. Concurrently, a Cooper pair enters the semiconductor. Fig. 6.2C illustrates the Andreev reflection process of an electron traveling from S_1 to S_2 and a hole traveling the opposite way. The hole in this case will reach the first superconductor and will also be reflected. However, now it becomes an electron with energy E′. When there is no bias, these energies are equivalent, $E = E'$. The reflection process can go on for a long duration, creating an infinite loop. This is sometimes called cross-Andreev reflection (CAR). The process is symmetric in that incoming holes are also reflected as outgoing electrons. There is symmetry for the positively charged hole Andreev reflection and the electron Andreev reflection. Particles are in a bound state in between semiconductors similar to quantum dots. The states form a discrete spectrum due to the confinement between the two semiconductors. These energy levels are the Fermi levels of energies of respective semiconductors.

The conditions for the formation of a bound state include the total phase in the one full transition to be a multiple of 2π. This means that an integral number of wavelengths must exist. Therefore, the phase difference is modeled as a combination of several phase differences. In particular, $2\pi = \phi_2 - \phi_1 + L(k - k') - 2\arccos(E/\Delta')$, where $\Delta = \phi_2 - \phi_1$ and $|\arccos(E/\Delta')| << L(k - k')$. Here, L is the length of the junction and k and k′ have to do with phase difference between electrons and holes. Also $\arccos(E/\Delta')$ is due to confinement energy gap. The gap is a region where there is a lack of density of states near the Fermi energy. This is also called a pseudogap. If there is symmetry between the hole and electron, then $k = k'$, $E_k = +- c\cos(\Delta/2)$. When this happens, this results in an energy spectrum of bound states. When there is a long junction L and k differs from k′, then this is the case where energy grows linearly, $E_n = h\nu L/2\,(2\pi\,(n + 1/2) +- \Delta)$. Previously implied, there are hole-like and electron-like Andreev bound states. For a hole-like bound state, a minus sign is used. However, the overall energy is zero.

The energy during Andreev reflection for the S_1 N S_2 transition, that is, for the semiconductor, metal, semiconductor transition, is given by $E = +- c\cos(\Delta/2)$. For an S_1 D S_2 configuration, a value τ is needed in the energy. It is a transmission value describing the tunneling effect, and $E = +- c(1 - \tau\sin^2(\Delta/2))^{1/2}$. There exists higher order Cooper pair tunneling, $I(\Delta) = I_c\sin(\Delta) + I_c'\sin(2\Delta) + \ldots$. The first term is for Cooper tunneling, the next term is for two Cooper pair tunneling, etc. There is also a Majorana fractional JJ effect; it provides energy $E = +- c'\cos(\Delta/2)$. This results in a 4π period in the JJ effect (Frolov, 2014). Also see Frolov (2019).

The second JJ equation is given by $2eV = h\,\partial(\Delta)/\partial t$. It involves the time dependency of JJ and shows the normal state of JJ. Above the critical current, there is still a changing phase. Using both JJ effects shows there is AC current, and therefore radiation is created. To see the radiation, microwaves are used to drive Shapiro steps (Kopnin, 2009). These steps appear in superconducting quantum interference and can be seen in a current versus

voltage graph similar to the one illustrated previously. The Josephson frequency is 2 e V/h. For any JJ, there is a Josephson energy; it is a free energy W stored within the junction. $W = \int I\ V\ dt = h/(2e) \int I\ d\Delta$. The supercurrent $I = 2e/h\ \partial\ (W)/\partial\ (\Delta)$, where $W = E\ (1 - \cos(\Delta))$.

By referring to Fig. 6.3A, there are three currents, one in each branch of the parallel circuit: The JJ supercurrent $I = I_c \sin(\Delta)$, which is nonlinear. The normal current $I_{nc} = h/(2eR)d\Delta/dt$, and the charge current $I_{cc} = C\ dV/dt = hC/(2e)\ d^2\Delta/dt^2$. The total current flowing through the circuit is often referred as an RSCJ model; it is $I\ (d^2\Delta/dt^2, d\Delta/dt, \Delta)$, and RSCJ stands for resistive superconductive capacitative junction. In any case, $I\ (d^2\Delta/dt^2, d\Delta/dt, \Delta) = I_{cc} + I_{nc} + I = hC/(2e)\ d^2\Delta/dt^2 + h/(2eR)d\Delta/dt + I_c \sin(\Delta)$. This equation is similar to the harmonic oscillator equation. However, as previously mentioned, the harmonic oscillator has energy levels all with equally distant energy levels. The energy levels for the RSCJ model given earlier have energy levels nonequally spaced. See Fig. 6.3B. In this diagram, the bottom two levels are indicated with kets, $|0>$ and $|1>$. In most quantum applications, the lowest two energy levels are most important. The bottom-most energy level is $|0>$, and it is referred to as the ground state. Above this state is the excited state, $|1>$. In this application, a cosine wave contains these energy levels.

For just a conventional LC circuit, the energy versus flux diagram has an energy-level separation always of equal distance. The distance between energy levels in this case is equal to $(LC)^{-1/2}$. Additionally, for linear equation of a typical LC circuit, the energy levels are described within a parabola-type boundary. In the RSCJ model illustrated in the figure, as previously mentioned, the nonequally distant energy levels lie within a cosine wave. Moreover, the energy levels have differences that become closer and closer for higher and higher energy levels. The reason for the nonequal distance between energy levels is to solve the problem of controlling qubits. With equal-distant energy levels, the states cannot be manipulated with precision using microwaves or lasers. Qubits tend to jump to higher and higher energy levels. To create a qubit, usually only the two bottom energy levels are employed. To obtain nonequally distant energy levels is the reason for the utilization of nonlinear elements, such as the JJ.

Use of a new material sandwiched between super-conductors leading to topological qubits is described in Zhu et al. (2022). Also, a rigorous account of the Andreev effect is described in Dolcini (2009).

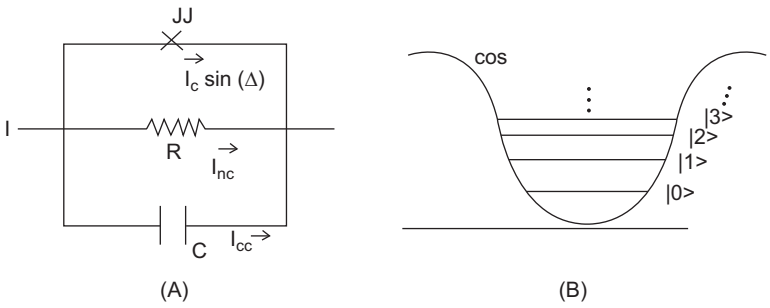


FIGURE 6.3 (A) Parallel Circuit, (B) Energy levels within cosine type boundary.

There are three types of Josephson qubits: charge qubits, flux qubits, and phase qubits, besides hybrid qubits. These qubits can be thought to be formed using a nonlinear resonator produced by Josephson inductance and the junction capacitance. The basic difference between these JJ qubits is the distinct circuitry needed for creation. In the charge qubits, two JJs are utilized. In the others, only a single JJ is employed. The energy levels, as usual, are not equally spaced and are described as parallel levels in a nonlinear potential curve, like the one illustrated in Fig. 6.3B. The nonlinear potential curve providing the border to the discrete energy levels is given in order for the charge qubits, the flux qubits, and the phase qubits. They are cubic, quartic, and cosine shaped, respectively. The latter shape is illustrated in Fig. 6.3B. Charge qubits are also known as a Cooper-pair box.

Single transmon qubits consist of two energy levels; these qubits are housed on a substrate that is a dialectic and also with a readout resonator. A laser usually controls an atom on the lowest two energy levels. As previously mentioned, there are several techniques for making a qubit. Transmon qubit is a superconducting charge qubit that is not sensitive to charge noise. TRANSMON stands for TRANSmissiON line shunted plasma oscillator qubit. One substrate with two plates with a nonlinear inductor JJ-type bridge in between. There is a positive charge on one and a negative charge on the other. The dimensions are in microns. The basic concept again is the transfer of Cooper pairs (Blais et al., 2020; Gambetta et al., 2017).

6.9 Quantum dots

Quantum dots (QDs) are nanocrystals of a semiconducting material. These are semiconductor qubits, but very different from JJ qubits. Quantum dot qubits are created when electrons or holes are enclosed in a static potential well within a semiconductor. This creates a quantized energy spectrum. Qubits have states corresponding to the spin of an electron. For instance, $|0>$ denotes spin up and $|1>$ indicates spin down. Superposition of these states $|v>$ is described as usual $|v> = a\ |0> + b\ |1>$, where a and b are complex numbers. Fabrication most recently involves a silicon, Si, substrate. A quantum well is also constructed using Si. On both sides of Si, a silicon geranium substrate is employed. Electric fields control the qubits consistent with scalable nanotechnology. QDs are also called artificial atoms.

In the beginning of QD fabrication, 1998, other materials were used in sandwiching different types of semiconductors, early on Ga and As were used. Two main problems exist with Ga and As atoms. They have finite nuclear spin, and there is hyperfine stochastic interaction, thus resulting in the qubit intrinsically de-cohere within 20 ns. This means, for instance, that the state in superposition $|v>$ might collapse to $|1>$ or $|0>$, on the average of 20 ns. Another problem is creating the precise localized oscillating fields to control a single qubit is difficult in these materials. A remedy is to use Si and Ge instead of Ga and As, even though the new substances have a more complex band structure. Again from the beginning, the objective of QD technology was to use the qubits in a gate-oriented quantum computer. This involves the Loss and DiVincenzo proposal for making a computer using a semiconducting material. Also see Section 6.5. Using Si and Ge results in compact high coherence and CMOS-compatible electric gates.

Originally also, the charge was used to determine the state, but as previously mentioned, again the de-coherence was problematic. De-coherence occurred due to a change in energy resulting in a phase change, leading to de-coherence. The result is that spin qubits are presently used in QDs. However, spin is converted to charge and charge is measured. In short, charge is indicative of the spin. Measuring charge is simple, but measuring spin directly is difficult. Spins are only weakly coupled to the environment (Loss, 1998). Two criteria for measuring the rate at which the environment interferes with qubits are the relaxation rate and the de-coherence rate. The reciprocal of the relaxation rate is the relaxation time T_1. It is the average time it takes for a spin qubit to keep its state, without a deliberate action to change the state. A spin up qubit should not convert to spin down or vice versa. In present-day devices, $T_1 = 1$ minute. Coherence time, $T_2 = 100$ microseconds, is the average time to keep superposition (Danon, 2021).

6.10 D-wave adiabatic quantum computers and computing

The actual D-wave AQC is highly shielded and is kept close to vacuum as well as near absolute zero, 0.015 K. The overall power needed is mainly for refrigeration. There exist grids of flux-type qubits within the low temperature region. Voltages and magnetic fields are used to control the qubits. Additionally, these sources have the capability to read the flux qubits. There is a lattice of superconducting loops, with each loop acting like a spin up or down particle. The programming parameters, described later, are converted to electrical currents, voltages, and magnetic fields. The electrical energies engage qubits in a superposition state. Controlled energies create tensor products of qubits employing the addition and scalar multiplication described within the Hilbert space structure. As such, the qubit spins are entangled.

Since the D-wave computer is an AQC; it only solves optimization-type problems. Accordingly, during the annealing process, the qubits search the feasible regions of the problem space. At the end of the annealing cycle, a ground state is obtained. After observing spin states, postprocessing may be performed. The process may be repeated thousands of times. The end product cannot tell if there is no solution or even if it is the global best solution. It only provides a solution.

Problem formulation is most often done using the quadratic unconstrained binary optimization (QUBO) model. Although other less mathematical models are also employed, the QUBO model either maximizes or minimizes an objective function of the affine form, $w = v'Qv + d$. Here, v is an n by 1 real-valued vector. The quantity Q is an n by n upper triangular matrix with real entries. The optimization does not involve d; it only involves $v'Qv$. In this representation, all tuples within v are idempotent, that is, $v_j^2 = v_j$. The annealing process results in an adiabatic search instead of brute force calculations on standard computers. The result might not be the optimal solution using AQC. It might be a suboptimal solution, but it will not take a week or more to acquire the solution.

Using quantum annealing, qubits tunnel through barriers trying to find the lowest energy state. The annealing cycle is about a microsecond in solving optimization-type problems. The objective function could be written as a minimization problem, namely as to minimize $w = \sum a_i v_i + \sum_{i<j} b_{j,k}\, v_j\, v_k$. The quantities a_i are located on the main diagonal

of Q and are called weights. The values $b_{j,k}$ are called strengths and lie on off diagonal entries of Q. The qubit ultimately settles into one of two final states: {0, 1}. The pair v_j v_k is called a coupler; it allows one qubit to influence another qubit in an entanglement. The values a_i and $b_{j,k}$ get mapped into voltages, currents, and magnetic fields in order to manipulate the qubits in discovering a global minimum.

Quantum tunneling D-wave chips start out with Hamiltonian in a ground state. The state has a minimum gap between it and other excited states. The gap is such that there is just enough energy so that it will not jump to a higher state. The process is performed very slowly. D-wave built a chip to do optimization with a nearest neighbor model. On the chip, the QUBO problem will map to a quadratic optimization problem. The architecture uses crossbar technology with integrated circuits. Next, an example is given to illustrate the affine objective function used by D-wave. It also emphasizes the idempotent operation used in the QUBO software (O'Malley et al., 2017).

Example 6.11:

For a two-dimensional example, using the notation involving the affine representation, $w = v'Qv + d$, where $v = (x\ y)'$, and Q is the upper triangular 2 by 2 matrix, given below, and followed by the 2 by 1 vector, Qv,

$$\begin{vmatrix} a & c \\ 0 & b \end{vmatrix} \quad \begin{vmatrix} ax + cy \\ by \end{vmatrix}.$$

The quantity $w = v'Qv = a\,x^2 + c\,x\,y + b\,y^2 + d$. But, since in QUBO power operations are idempotent, this means that $x^2 = x$ and $y^2 = y$. Therefore, it follows that $w = a\,x + b\,y + c\,x\,y + d$, and this is called the spectrum in QUBO.#

Example 6.12:

Here is a simple example for values x and y in {0, 1}. The objective is to find $z = (x + y) = 0 \bmod 2$. This is the situation where x and y are equal, and is the equivalence operation in logic. A truth table will be constructed. In the Spectrum column, when the logical operation is true, a zero is placed in this column. Zero is indicative of the objective to obtain the lowest energy, $0 < 1$.

x	y	Spectrum = w	
0	0	0	Lowest energy
0	1	1	
1	0	1	
1	1	0	Lowest energy

A glance at the truth table above renders the solution $y = x$. Utilizing the results from the previous example, the affine mapping can be constructed.

The spectrum value in this case is found from the previous example, $w = a\,x + b\,y + c\,x\,y + d$. The quantities Q and d must be found. To do this, go row by row in the truth table. Substitute the values of x and y and also use the value of the spectrum w. For the

top row, all values, x, y, and w, are zero. Accordingly, 0 = w = a 0 + b 0 + c 0 yields d = 0. From the second row, 1 = w = a 0 + b 1 + c 0 shows b = 1. Using the third row 1 = w = a 1 + c 0 implies that a = 1. Finally, the bottom row 0 = w = a 1 + b 1 + c 1 1 results in c = −2. Accordingly, w = x + y − 2 x y. As a consequence, d = 0 and Q =

$$\begin{vmatrix} 1 & -2 \\ 0 & 1 \end{vmatrix}.\#$$

When using an AQC, an objective function must be specified, which provides the energy level for some variables. When these variables attain that energy level, then a solution occurs. It corresponds to a low energy state or the lowest energy state with high probability. The final state of the qubits represents the solution.

Programming model for D-wave:

Quantum bit, v_i: This value takes part in the annealing process and reaches a final state in {0, 1}. The quantum bit usually undergoes interference using quantum superposition.
A coupler, v_i, v_j: This pair allows a single qubit to influence another qubit and thereby produces entanglement.
A weight a_i: The weight is a real scalar assigned to each qubit. It influences a qubit to collapse into a final state. These quantities are diagonal elements in the matrix Q.
Strength, b_{ij}: This is a real scalar assigned to each coupler. It influences the control of one qubit over another. These are off-diagonal elements of Q.
Objective, Obj cost function, is minimized during annealing cycle. This is w.

6.11 Adiabatic theorem

An adiabatic process is a slow and gradual changing event in which quantum systems avoid jumping states or changing quantum numbers. The adiabatic invariant I is the quotient of the energy divided by the frequency, E/w. As an instance of the adiabatic process, consider the harmonic oscillator. In Example 6.1, the energy levels were stated to be $E_n = hw\ (2n + 1)/2$. Because the adiabatic invariant in the harmonic oscillator situation involves integers, a continuous change is impossible. The process must take discrete or quantum jumps in changing. A large number of other quantum systems have a discrete spectrum, and they have corresponding energy levels that jump in value. In these cases, there is an avoidance to change particularly under slow varying conditions. Another primary example is the Bohr-Sommerfield (Hall, 2013) quantum model of a miniaturized solar system. In this case, quantization conditions exist for which quantum numbers do not want to change under slow varying conditions. A most readable explanation of the adiabatic process and its relationship with the Berry's phase, introduced later, are provided in Zwanziger et al. (1990).

The adiabatic theorem described herein is given formally. It follows the excellent presentation of Zwiebach (2018). Specifically, it is shown that the error in the difference between the true Schrödinger solution rendering the state, Ψ(t), and an instantaneous eigenstate, ψ(t), is of order O (1/T). This relationship holds during the adiabatic process, having duration T. Symbolically, $\|\Psi(t) - \psi(t)\|$ is of order O (1/T), for t in [0, T]. The length

of time T is the duration of the adiabatic process. An instantaneous eigenstate $\psi(t)$ of the Hamiltonian is a wave function $\psi(t)$, having the property that $H(t)\ |\psi(t)> = E(t)\ |\psi(t)>$. Here, H is the Hamiltonian, and E(t) is the energy eigenvalue. As previously mentioned, it is difficult to make quantum transitions in slowly varying processes since energy levels are usually discrete. A formal account of the adiabatic theorem follows.

Inspired by the time-independent Schrödinger equation, $H(t)\ |\psi_k(t)> = E_k(t)\ |\psi_k(t)>$, $k = 1, 2, \ldots$. The adiabatic condition provides an approximate solution to the time-dependent Schrödinger equation. This solution involves the instantaneous eigenvalue and the eigenstate equation involving the Hamiltonian. In this situation, it has to be that during the process the energy changes slowly. It is also assumed that all energy levels are far apart, that is, $E_k(t)$ is well separated from all other discrete energy levels. Moreover, say that the following inequalities hold between adjacent energy levels, $E_{k-1}(t) < E_k(t) < E_{k+1}(t)$. The adiabatic theorem provides an approximation for the true state $|\Psi_k(t)>$. It says that $|\Psi_k(t)>$ is about equal to $e^{i\theta_k}e^{i\gamma_k}\ |\psi_k(t)>$. Another required condition is the equality of initial conditions, that is, $|\Psi_k(0)> = |\psi_k(0)>$. Here, the approximate solution to Schrödinger equation at time zero equals the instantaneous energy eigenstate solution at time zero. The exponentials in the approximation are described in the next paragraph, and the following analysis is only formal.

The real-valued quantity, θ_k is the dynamical phase for energy eigenstates. The dynamic phase is given by $\theta_k(t) = -1/h\int_0^t E_k(t')dt'$. The other real-valued quantity is the Berry phase v_k; it too is real valued. The Berry phase is given by the inner product of the approximate wave function, and it's derivative. This inner product is $v_k = i < \psi_k(t), d\psi_k(t)/dt>$. Also, let $\gamma_k(t) = \int_0^t v_k(t')dt'$. Next, a quantity similar to the Berry phase will be investigated. It is called the coupling term, and it involves distinct energy states. The objective is to see how the coupling term relates to the Hamiltonian, as well as to determine to what extent the coupling term influences the adiabatic process.

To see how the coupling term $<\psi_n(t), d\psi_k(t)/dt>$ relates to the Hamiltonian, first consider the time instantaneous eigenfunction condition for k different from n, $H(t)\ |\psi_k(t)> = E_k(t)\ |\psi_k(t)>$. Differentiating this eigenfunction equation results in $dH/dt\ |\psi_k> + H\ |d\psi_k/dt> = dE_k/dt\ |\psi_k> + E_k|d\psi_k/dt>$. Here, the last identity just uses the eigenfunction equation twice. Next, apply the bra operation, $<\psi_n(t)|$, on the left of both sides of the last identity. This operation sandwiches the entries in the previous equation. Doing this gives $<\psi_n(t)\ |dH/dt\ |\psi_k> + <\psi_n(t)\ |H\ |d\psi_k/dt> = <\psi_n(t)\ |dE_k/dt|\psi_k> + <\psi_n(t)\ |E_k|d\psi_k/dt>$. Employ the eigenfunction equation, $H\ |\psi_n(t)> = E_n\ |\psi_n(t)$, again and convert it into an equation involving bras. In order to do this, use conjugating and remember that the scalar field is real; as a result, $<\psi_n(t)\ |H = E_n\ <\psi_n(t)\ |$. Also, note that $<\psi_n(t)\ |dE_k/dt = 0$, when k and n differ. Putting this all together gives $<\psi_n(t)\ |dH/dt\ |\psi_k> + E_n < \psi(t)\ |d\psi_k/dt> = E_k < \psi_n(t)\ |d\psi_k/dt>$. Notice that the last two terms have an inner product in common, which is exactly the coupling term. Solving for this term shows that $<\psi_n(t)\ |d\psi_k/dt> = <\psi_n(t)\ |dH/dt\ |\psi_k>/(E_k - E_n)$. The final result is $<\psi_n(t)\ |d\psi_k/dt> = (dH/dt)n,k/(E_k - E_n)$, where as mentioned before that E_k and E*n* are different. The quantity, $(dH/dt)^{n,k}$, is the n,k entry in the derivative of the Hamiltonian matrix. The conclusion is that the coupling term can ruin the adiabatic approximation. This is the case whenever the n,k term in the derivative of the Hamiltonian is small. The following sequence of examples follows (Zwiebach, 2018).

Example 6.13:

This is an example illustrating the error in the adiabatic approximation. It shows for time T increasing, the error decreases as 1/T. Consider a Hamiltonian $H(t) = H_0$ for $t < 0$, $H_0 + H_1\ t/T$, for t in [0, T], and $H_0 + H_1$, for $t > T$. Here, $0 < T$ and $0 < H_0 < H_1$. See Fig. 6.4A. For t in [0, T], $dH/dt = H_1/T$. All errors in the approximation vanish like 1/T.#

The Landau-Zener transitions describe adiabatic as well as possible nonadiabatic changes of eigenfunctions, $\psi_1(x,R)$ and $\psi_2(x,R)$. They represent two distinct electronic arrangements of a molecule with a fixed nuclei such that they are separated at a distance R apart. In Fig. 6.4B, there is an illustration of the energies associated with the corresponding eigenfunctions. For instance, in this diagram, $E_1(R)$ might be the energy of an eigenstate wave in ground state and $E_2(R)$ the energy of an excited state. In this diagram, both energy functions with nonpolar and polar regions appear as a function of R on the abscissa. Here, it assumed that the states change characteristics at $R = R_0$. For the eigenfunction $\psi_1(x,R)$, it transitions from polar to nonpolar at R_0. For the second eigenfunction $\Psi_2(x,R)$, the opposite occurs. If R increases quickly, then $\psi_1(x,R) \rightarrow \psi_2(x,R)$. This is nonadiabatic change occures, but if R changes slowly then the eigenfunctions will remain on their original trajectories.

The Hamiltonian equations are $H(R)\ \psi_i(x,R) = E_i(R)\ \psi_i(x,R)$, i = 1, 2. If this is solved for all R, then the instantaneous energy eigenstates are found as described in the adiabatic theorem. Assume that R is a function of time t, then so is the Hamiltonian, H(R(t)). The corresponding eigenfunction equations become $H(R(t))\ \psi_i(x,R(t)) = E_i(R(t))\ \psi_i(x,R(t))$, i = 1, 2. If the top equation is solved for all R, then the time-dependent equation also holds for all t. So the instantaneous energy eigenstates and eigenvalues are known from the top equation given above.

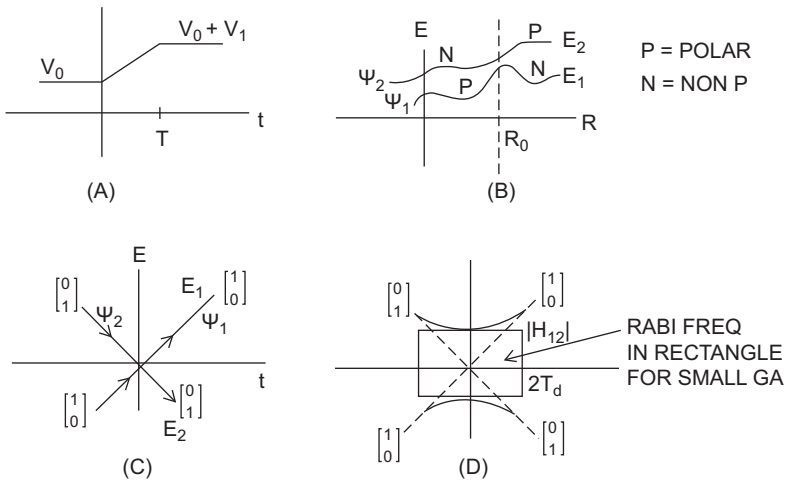


FIGURE 6.4 Adiabatic process following Zwiebach, (A) Hamiltonian, (B) Energy Separation, (C) Path Crossing Hamiltonian, (D) Non Crossing Paths.

Example 6.14:

This example illustrates a Hamiltonian H, for which the states will stay on their trajectory even though their paths cross. Fig. 6.4C illustrates a plot of the instantaneous eigenvectors in a time vs. energy diagram. Let H =

$$\begin{matrix} |ct/2 & 0| \\ |0 & -ct/2|, \end{matrix}$$

where c is a nonnegative scalar with eigenvectors (1 0)′ and (0 1)′. In the Fig. 6.4C, these eigenvectors lie on 45 degree lines. The corresponding energy values are $E_1 = ct/2$ and $E_2 = -ct/2$. Notice that $\Psi_1(t) = e^{-i/h\int_0^t E_1(t')dt'} |1> = e^{-ict^2/(4h)} |1>$ and $\Psi_2(t) = e^{i/h\int_0^t E_2(t')dt'} |2> = e^{ict^2/(4h)} |2>$. These are the instantaneous eigenfunction solutions. However, in this case they also are the exact solution to the Schrödinger equations. That is, $ih\partial\Psi_1(t)/\partial t = H\ \psi_1(t)$ and $ih\psi_2(t)/\partial t = H\ \psi_2(t)$; direct substitution illustrates this. For instance, for the first eigen vector, $ih\partial e^{-ict^2/(4h)}/\partial t = ih\,(-2ict/(4\,h)) = ct/2$.#

In the example mentioned earlier, the trajectories for ψ_1 stay on (1 0)′ forever. There is no influence at the origin even though the paths cross. The same is true for ψ_2. A different story occurs in the next example.

Example 6.15:

This time, assume that the diagonal terms of the Hamiltonian matrix are the same, but there is correlation in the form of complex-valued off-diagonal entries. Notice, however, that the matrix is still self-adjoint. So, H = H* =

$$\begin{matrix} |ct/2 & H_{12}| \\ |H_{12}^* & -ct/2| \end{matrix}$$

The trajectories are again illustrated in a time vs. energy diagram, this time in Fig. 6.4D. In this case, the paths do not cross; however, they can get arbitrarily close to each. This would be the case for H_{12} to go to zero. The interesting cases are if there exists a gap, and how large it is. To analyze this situation, refer to Fig. 5.4D. There are four points of interest. These are where the rectangle intersects the abscissa and the ordinate. At the ordinate, the rectangle is tangent to the parabola-type figure. The height is $+ -\ |H_{12}\ |$; these are also the energy eigenvalues at time 0. On the abscissa, the rectangle intersects at $+-2\tau_d$. The shape of this rectangle governs the adiabatic or nonadiabatic process. That is, these two parameters $|H_{12}\ |$ and τ_d determine the conditions of whether the instantaneous energy eigenstates change, polar to nonpolar, namely (0 1)→(1 0), or vice versa. When the gap is large, the adiabatic process prevails and there is no change of trajectories. When the gap is small, say near zero, then the Rabi frequency $w_{12} = |H_{12}\ |/h$ occurs in the rectangle, and a nonadiabatic process happens. The probability of a nonadiabatic transition is calculated to be $P = e^{-2\pi w_{12}\tau_d} = e^{-2\pi H_{12}^2/(hc)}$. The solution involves hypergeometric functions (Lahiri, 2006). Appendix A.7 describes the hypergeometric functions in terms of Sturm-Liouville differential equation.#

Berry phase, v_k, is understood as a geometrical phase, whereas θ_k is as usual a time-varying function. It is understood that the Berry phase describes a geometric path and only depends on evolution of space, but not on time. As a consequence, this leads to the Berry connection $A_k(R) = i < \Psi_k(R) \mid \Delta_R \mid \Psi_k(R) >$. Here, R is a vector in configuration space with N components, namely R is a vector in R^N and Δ is the gradient operation. Eigenvectors are not unique; any nonzero scalar times an eigenvector is also an eigenvector. However, states are of norm one. This implies that $\Psi_k(R)' = e^{-ib(R)}\Psi_k(R)$ is also an eigenvector provided that the quantity b(R) is real valued. Use the Berry connection with respect to the new eigenstate. Substituting in and using the gradient operation results in $A_k(R)' = i < \Psi_k(R) e^{ib(R)} \mid \Delta_R \mid e^{-ib(R)} \Psi_k(R) > = A_k(R) + \Delta_R b(R)$. This is the gauge transform under gauge connections. Intuitively, a gauge transform changes the scale, and it is the gauge connection, which provides the compensation between the original and new scale. Gauge transformation applied to the Berry phase is not gauge invariant. Because $\gamma_k(t) = \int_0^t v_k(t')dt'$,it can also be written only involving the geometric path in R^N. In this case, $= \int_\Gamma A_k(R)dR$. Using the Berry connection with the new eigenvector gives $\gamma_k(t)' = \int_\Gamma A_k(R)'dR = \int_\Gamma A_k(R)dR + \int_\Gamma \Delta_R d(R)$. This shows that the Berry phase cannot be observed unless the motion in the configuration space is a loop. Consequently, only at multiples of 2π, the Berry connection is gauge invariant (Zwanziger et al., 1990).

Reference

Aharonov, D. et al., 2014. Adiabatic quantum computation is equivalent to standard quantum computation. In: Proc of FOCS-04, 45th Annual IEEE Symposium on Foundations of Computer Science.

Blais, A., et al., 2020. Circuit quantum electrodynamics. arXiv:2005, 12667.

Blinov, B., et al., 2004. Quantum computing with trapped ion hyperfine qubits. QIP vol. 3, no. 1–5.

Carson, B., 2014. Potential well, Physics and Astronomy dept., Carthage.

Childs, A., et al., 2001a. Adiabatic quantum computation review. arXiv link.

Childs, A., Farhi, E., Preskill, J., 2001b. Robustness of adaibatic quantum computation. Phys. Rev. A 65, 012322.

Danon, J., 2021. Solid state spin qubits, Gemini Center for quantum. NTNU. lcrostaic gates Quasi 1D constrictions.

DiVincenzo, D., 2013. Quantum computers, at TEDxEutropis.

Dolcini, F. 2009. Andreev reflection. In: Lecture Notes for XXIII Physics GradDays. Heidelberg.

Finke, 2021. The Qunnect Company reported on demand electromagnetically induced transparency quantum type memory Storage networking industry association, quantum computing report, SDC.

Frolov, 2014. Quantum Transport Lecture 12 Spin Qubits.

Frolov, S., 2019. Lange Lecture: "Status of the Search forMajorana Fermions in Semiconductor Nanowires". U. Pittsburgh.

Gambetta, J., et al., 2017. Building logical qubits in a superconducting quantum computing system. Npj Quantum Inf. 3 (1), 2.

Hall, B., 2013. 978-1-4614-7115-8 Quantum Theory for Mathematicians. GTM Springer.

Hui, J., 2019. QC-Hoot build a quantum computer with trapped ions Medium?

IonQ Staff, 2023. IonQ Forte: The firest software configurable quantum computer. Internet .

Kopnin, N., 2009. Introduction to the theory of superconductivity, Cryocourse.

Krauss, B., 2013. Topics in quantum informationIn: DiVincenzo, D. (Ed.), Quantum information Processing, Lecture Notes of the 44th IFF Spring School.

Lahiri, S., 2006. A sub-gaussian Berry theorem Report no. ISU-Stat-2004-21.

Mattel, N., 2021. Demonstration of Rag-flops with Ytterbium 171 trapped ion qubits. IJSRP 11 (7).

O'Malley, D., Vesselinvov, V., Alexandrov, B., Alexandrov, L., 2017. Nonnegative/binary matrix factorization with a D-wave quantum annealer. arXiv:1704-01605 .

Ortu, A., et al., 2022. Storage of photonic time bin qubits for up to 20 ms in a rare earth doped crystal. Npc Quantum Inf. 8, 29.
Ruster, T., et al., 2016. A long lived Zeeman trapped ion qubit, Springer Link, rapid communication. Appl. Phys. B 122, article number 254.
Scerri, E., 2013. The trouble with the Aufbau principle. Education in Chemistry 50 (n6), Royal Society of Chemistry.
Simon, M., et al., 1995. Spin stabilized magnetic levitation. AJP 65 (4), 286–292.
Stas, P., et al., 2022. Robust multi-qubit quantum node with integrated error detection. Science . Available from: https://doi.org/10.1126/science.add9771.
Wendin, G., 2017. Quantum information processing with superconducting circuits: a review. Rep. Prog. Phys. 80 (10), 106001.
Zhang, J., et al., 2018. Realizing an adiabatic quantum search algorithm with shortcuts to adiabaticity in an ion trap system. Phys. Rev. A 98 (5), 052323.
Zhu, Z., et al., 2022. Phase tuning of multiple Andreev reflections of Dirac fermions and the Josephson super current in Al-$MoTe_2$-Al junctions. PNAS 119 (28).
Zwanziger, J., et al., 1990. Berry's Phase. Lawrence Berkeley Labs, UC of Berkely.
Zwiebach, B., 2017. Quantum Physics : Adiabatic Approximation. MIT OpenCourseWare.
Zwiebach, B., 2018. MIT Courseware.

Further reading

Farhi, 2001. Framework of adiabatic quantum optimization. Science, arrive.org.
King, Theory of quantum annealing Google TechTalks (Nishimori, 2015). arXiv:2202.05847
Liu, J., et al., 2018. Adiabatic quantum computation applied to deep learning networks. Entropy 20 (5), 380.

This page intentionally left blank

CHAPTER

7

Operators on Hilbert space

7.1 Linear operators, a MSA view

Among the most fundamental operators employed in quantum technologies are the linear maps. All operators in this document will be linear unless specified otherwise. They satisfy the linearity condition mentioned later. Numerous examples of these maps have been seen before; they include linear maps that are linear in both arguments. Also included are multilinear operations and tensor maps. Both of which are linear in all their operands. However, conjugation is not linear, and so the inner product is also nonlinear; it is only linear in the second argument. This type of nonlinearity, being conjugate linear only in the first argument and linear in the second argument, is called sesquilinear. The set of all linear maps involving vector spaces V and W over the complex field C or R itself forms a vector space. In this description utilizing the MSA, it follows that there are four sorts: V-VECTOR, W-VECTOR, C-SCALAR, and L-VECTOR all using suggestive notation. In particular, the last sort denotes the sort of all linear maps, L (V, W) from vector space V into vector space W.

All the signature sets for the vector space structures and field structure are as before and will not be repeated. However, a new signature set containing a unary operator name LIN exists and is such that:

$$\text{LIN} : \text{V} - \text{VECTOR} \rightarrow \text{W} - \text{VECTOR}$$

Additionally, to be linear, there is a single constraint that must be satisfied. It involves vector spaces with sorts V-VECTOR and W-VECTOR along with corresponding scalar multiplication and vector addition operational names. To see this, first, replace all names with suggestive symbols, that is,

LIN by T
V-VECTOR by u, v
Vv-ADD by $+$
Sv-MULT by $\cdot$
W-VECTOR by w
Vw-ADD by $+_2$
Sw-MULT by $\cdot_2$
C-SCALAR by a

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00020-5

The linearity condition is as follows:

1) Linear: $T(a \cdot u + v) = a \cdot_2 T(u) +_2 T(v)$

As previously mentioned, the set of all linear maps from V to W is denoted by the name L (V, W). It itself becomes a vector space when point-wise addition and scalar multiplication are defined in L (V, W). Additionally, for this fourth sort, that is, L-VECTOR also needed is the corresponding equational identity to ensure that L (V, W) is a true vector space. As before, the signature sets associated with L-VECTOR involve scalar multiplication, SL-MULT, and vector addition VL-ADD.

So represent as follows:

L-VECTOR by S, T
VL-ADD by $+_3$
SL-MULT by $\cdot_3$

Actually, S and T mentioned earlier are themselves linear mappings from vector space V into vector space W.

The vector space condition needed for maps in L (V, W) is as follows:

2) Linearity: $((a \cdot_3 S) +_3 T)(v) = a \cdot_2 (S(v)) +_2 T(v)$

This vector space L (V, W) is often also denoted by Hom (V, W), which is the space of all homomorphisms from V To W. Carrier sets could be defined, for instance, when bases are chosen. In this case, the linear maps can be represented by matrices in finite dimensions. Here, the usual matrix and vector operations relate to the operator names within signature sets.

Use vector spaces V and W and the operator T in L (V, W). If the condition that u and v are not equal in V implies that T (u) is not equal to T(v), then the mapping from V into W is said to be one-to-one or injective. Also when the codomain of T is all of W, then the mapping is called onto or surjective. When both of these are true, there exists an inverse map: T^{-1}: W $\rightarrow$ V, and it too is linear. The spaces W and V are also isomorphic under this mapping.

If V = W, the vector space of all linear maps becomes an algebra when the composition of linear maps is defined. The resulting structure is denoted by End (V) and is called the set of endomorphisms. These transformations have a domain equal to their codomain, thus allowing the composition operation to be performed. Here S and T have the same domain, with a domain equal to its codomain, that is, they map V into V. The composition operation C[S] is itself a linear operator defined by C[S] (T) = S (T(.)). That is, it creates a composition of maps for S and T, which becomes a closure, multiplication-type operation, but not necessarily a commutative operation. However, when identities (1), (2), (3), and (4) of Section 3.1 hold, then End (V) becomes a unital associative algebra. The polyadic graph for a unital associative algebra is illustrated in Fig. 3.1. It is assumed that all the operators in End (V) satisfy the closure operations illustrated in this figure. In particular, V-ONE corresponds to the identity operator I. In terms of End(V), the equational identities that hold for a unital associative algebra involve the following elements:

In End(V) are S, T, and P, as well as the identity map, I.

In C are scalars a and b

Also, instead of first using the operator T and then using S on vectors in V, as described earlier, C[S] (T) = S (T(.)), use instead the notation, S • T. This notation is employed in the equational identities:

1) Associative: P • (S • T) = (P • S) • T
2) Distributive: (a P + b S) • T = a P • T + b S • T
3) Distributive: T • (a P + b S) = a T • P + b T • S
4) Identity: I • T = T • I = T

Example 7.1:

The multiplicative notation applied in place of composition could be very misleading, particularly in infinite-dimensional vector spaces. For instance, for S and T in End(V), the quantity $S = T \cdot T - 4\,T - 5 = T^2 - 4\,T - 5 \cdot I$, when factoring, or when using the FOIL rule, there needs to be proper justification. From an operational calculus point of view, S = (T +1) (T − 5). However, it is not known how the last expression is rigorously obtained. To understand the factoring procedures, write $S = T \cdot T - 4\,T - 5 \cdot I = T \cdot T - 5\,T + T - 5$. This results in just using vector space operations. Then by using the left-hand distributive law, number (3), it follows that S = T • (T − 5) + 1 (T − 5). Finally, using the right-hand distributive law, number (2) mentioned earlier, results in S = (T +1) (T − 5).#

A single element T in End(V) is called an automorphism of V if it is one-to-one and onto, that is, it is invertible. So, $T^{-1} \cdot T = T \cdot T^{-1} = I$. The set of all automorphisms of V constitutes an instance of the general linear group and is symbolized by GL. This group will be described in a later chapter involving Lie groups. In finite dimensions, End (V) is isomorphic to the space of n by n complex invertible matrices.

An important application of linearity in the transform T applies to some Hilbert spaces. This property has to do with the sesquilinear inner product. A most fundamental result in any complex Hilbert space H is that whenever <v, Tv> = 0 for all v in H, then T = 0. This follows because for any u in H then <(u + v), T (u + v)> = 0 and so <v, Tu> + <u, Tv> = 0. Implying that <v, Tu> = − <u, Tv>. Repeating the process this time using imaginary values also gives <(u+ iv), T (u +iv)> = 0, and <iv, Tu>+ <u, Tiv> = 0. This means that − <v, Tu> + <u, Tv> = 0 or <v, Tu> = <u, Tv>, and so <u, Tv> = 0 for all u, v in H, which implies T = 0. For a real Hilbert space, the result is not true, as the following example will illustrate.

Example 7.2:

Let the carrier set for the Hilbert space H be $H = R^2$, and use the linear operator T, given by the matrix:

$$\begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix}.$$

Then, for any vector v = (a b)′, in H it follows that T (v) = (−b a)′, and the inner product in H is the usual dot product, (−b a) (a b)′ = < v, T(v)> = 0, but T is not zero. #

Another illustration of the utility of a linear operator A in a Hilbert space involves the numerical range W. The numerical range is very important in characterizing the spectrum of operators. It is used in quantum computing to find rough estimates of eigenvalues. The numerical range is defined as W (A) = {< v, A v>, such that $\|v\| = 1$, v in the domain of A}. The Toeplitz-Hausdorff theorem shows that the subset W(A) of the complex plane is always convex (Gustafson, 1970).

Example 7.3:

Consider $H = C^2$; the objective is to obtain information about the numerical range of a linear operator given as a 2 by 2 matrix, A =

$$\begin{vmatrix} 2 & i \\ -i & 1 \end{vmatrix}.$$

A unit vector in H will first be substituted into W (A) = < v, A v>, just to get an idea of what the numerical range provides. So using v = (i 0)′, then A v = (2i 1)′ and W (A) = < |q>, A| |q> > = (−2i 1) (i 0)′ = 2. Similarly, if w = (0 1)′, then A w = (i 1)′ and <w, A w> = (0 1) (i 1)′ = 1.

To obtain a more general idea of the numerical range, consider the normalized qubit | q>, given as a 2 by 1 column vector, |q> = (cos (t/2) e^{ip} sin (t/2))′ with t in [0, π] and p in [0, 2π). Then, the 2 by 1 vector A | q > = (2 cos (t/2) + i e^{ip} sin (t/2) − i cos (t/2) + e^{ip} sin (t/2))′. The inner product <A | q>, |q>> = 2 $\cos^2$ (t/2) − i e^{-ip} sin (t/2) cos (t/2) + i cos (t/2) e^{ip} sin (t/2) + $\sin^2$ (t/2). This quantity could be simplified as W (|q>) = 2 $\cos^2$ (t/2) + $\sin^2$ (t/2) + 2 sin (t/2) cos (t/2) sin (p). W (|q>) = 1 + $\cos^2$ (t/2) + sin (t) sin (p).

The subset of the complex plane can be found corresponding to W (A) by plotting the locus of points for <A|q>, |q> as t and p vary throughout their domains. Recall that | q> given earlier is the most general qubit in H except for arbitrary phase. However, if used, the global phase would cancel out when using the formula for W (A).

For any complex n by n matrix A, W (A) = {v* A v/(v * v), v nonzero in C}, where v* is the conjugate transpose of v. This is the same formula as mentioned earlier. Additionally, an important attribute of the numerical range is the inner numerical radius, r (A). This quantity is the shortest distance from the origin to the boundary of W (A). That is, r (A) = min {|z|, such that z is on the boundary of W (A)}. Using math lab simulation, for this example, r (A) = 0.382. The actual numerical radius is the farthest distance a point z in W (A) is from the origin. That is, R (A) = max {|z| such that z is in W (A)}. Again by simulation, R (A) = 2.618. In the example mentioned earlier, since the numerical range is real, it follows that W (A) = [0.382, 2.618]. These values are found again in a subsequent chapter.#

Two important linear operators in Hilbert space H follow. An accretive operator A is such that the real part of its numerical range is nonnegative. That is, Re (<v, A v>) is greater than or equal to zero, for all v in the domain of A. As an instance of an accretive operation, see the previous example, since W (A) itself is nonnegative. An operator A is said to be dissipative when -A is accretive. These operators are important in the derivation of Stone's theorem and again in the Lumer-Phillips theorem, described in later chapters.

7.2 Closed operators in Hilbert spaces

Consider the operator T, which is always linear, unless specified otherwise. Say that T is in L (D, H2). So T: D → H2, where D is a subspace of H1 and H2, which are both Hilbert spaces. The operator T is said to be a closed operator, which means that if vn in D converges to v and given Tvn converges to w in H2, then v is in D and Tv = w. The subspace D is the domain of the operator and is referred to as dom(T). So, for instance, in a field structure such as all the complex numbers C, dom(1/z) = C − {0 + 0i}. An equivalent definition for an operator to be closed invokes the graph of T. Here, the column vector is a two tuple; (vn Tvn)′ → (v w)′ implies that (v w)′ is in the graph of T.

A stronger condition for an operator to be closed means that it is sequentially continuous. That is, for every vn in H1 converging to v, Tvn converges to w in H2 and Tv = w. That is, the operator T is very often only defined on a subspace of a subset of H1. When it is defined on all of H1 and it is closed, then it is also sequentially continuous. This is a consequence of the closed graph theorem, which actually holds in Banach spaces (Rudin, 1991), also see Appendix A.6. In Euclidean space R^n and f : D → R^n, where D is a subset of R^n, then f is continuous at a point v in D iff f(vn) → f(v) for all sequences vn in D such that vn → v. So sequential continuity does imply continuity under these conditions.

Example 7.4:

Let T: R → R, where T (0) = T0 = 0 and elsewhere T (t) = 1/t, T is not continuous on R. However, it is a closed operator. This follows by taking any sequence vn in R, which does converge to v in R. Surely, + − ∞ are not the limits of any such converging sequence. Any point v, in R other than zero, could be such that vn → v. Accordingly, it follows that automatically Tvn → w = 1/v. This implies sequential continuity at all points in R − {0}. Since the origin is an isolated point in the graph, this implies that T is a closed operator. Note that T is not sequentially continuous at the origin. The sequence of all zeros does converge to zero. So does the sequence T0, T0, . . .; it also converges to zero. However, the sequence 1, 1/2, 1/3 converges to zero, but T1, T1/2, T1/3, . . . = 1, 2, 3, . . . goes to ∞.#

Additionally, since the convergence only involves the induced norm from the Hilbert space, T is usually defined on a normed vector space such as a Banach space. These extensions to the definition mentioned earlier are important in infinite-dimensional vector spaces and will be presented in a later chapter. Finally, the sequentially continuous condition mentioned earlier implies continuity and therefore boundedness. These relationships are described in the next section along with several proofs.

7.3 Bounded operators

Perhaps, the most important linear operators in quantum are the bounded operators from a Hilbert space into another Hilbert space. This is true even though position and momentum operators are unbounded. Let T be in L (H1, H2), where H1 and H2 are

Hilbert spaces and T: H1 → H2. Then T is said to be bounded that means that there exists M > 0, such that for all v in H1 the norm inequality or Lipschitz condition holds:

1) Lipschitz Condition: $\|T(v)\|$ is less than or equal to M $\|v\|$.
Recall that $<T(v), T(v)> = \|T(v)\|^2$. The infimum of all such values of M is called the operator norm of T and is denoted by $\|T\|$. An equivalent condition to be bounded is that the linear operator is sequentially continuous; in this case:

2) Sequential Continuity: For every sequence, let v, v_n in H1, n = 0, 1, ... if $v_n \to v$, then $T(v_n) \to T(v)$.

Another formula for finding $\|T\|$ whenever T is bounded is to take the supremum of $\|T(v)\|/\|v\|$ for v nonzero. Equivalently, the supremum could be taken of $\|T(v)|$ for $\|v\| = 1$. To see this, using $\sup\|T(v)\|/\|v\| = c$, for v nonzero, then when $\|v\| = 1$, this sup $\|T(v)\|$ also equals c. Going the other way, if sup $\|T(v)\|/\|v\|$, for $\|v\| = 1$, equals c, then choose any nonzero vector w. Since v is of norm one, let v = a w. Substituting v into the norm expression, sup $\|T(v)\|/\|v\| = |a|$ sup $\|T(w)\|/(|a|\ \|w\|) = |a|/|a|$ sup $\|T(w)\|/\|w\| = \sup\|T(w)\|/\|w\| = c$.

The norm is called the operator norm, and it is useful in proving the triangle inequality. This inequality is the key result in showing that the vector space of all bounded operators is a normed vector space. To see this, let $\|v\| = 1$, where v is in H1, and S and T are bounded operators from H1 to H2. Then, $\|S + T\| = \sup\|(S + T)(v)\| = \sup\|S(v) + T(v)\|$; this follows from linearity. Now the latter supremum is the norm of a sum of vectors in the Hilbert space H2, and the triangle inequality holds there, from the definition of the inner product. So, $\|S + T\|$ is less than or equal to sup $[\|S(v)\| + \|T(v)\|]$, which is less than or equal to sup $\|S(v)\|$ + sup $\|T(v)\| = \|S\| + \|T|$, since $\|v\| = 1|$.

In a trivial vector space consisting only of V-ZERO, the norm of any operator is zero. All the following vector spaces in the text will be assumed to not be V-ZERO. The set of all bounded operators from H1 to H2 is denoted by B (H1, H2); moreover, it is a Banach space. This means every Cauchy sequence of operators converges, that is, this normed vector space is complete. Whenever the Hilbert spaces are of finite dimension, all linear operators are bounded. Additionally, matrix operations can be employed in manipulating elements from the corresponding carrier sets. An example is provided below.

Example 7.5:

Let the Hilbert space be $H = C^2$, and consider the operator T: H → H where T is given by the 2 x 2 complex matrix:

$$\begin{matrix} |2 & i| \\ |0 & 1|. \end{matrix}$$

The objective is to find the norm $\|T\|$ using sup $\|T(v)\|$ for $\|v\| = 1$. The vector of the unit norm is, as given before, defined by the 2 by 1 column vector describing the qubit, $v = |q> = (\cos(t/2)\ \ e^{ip} \sin(t/2))'$. Here the prime is the transpose. Additionally, the value t is in [0, π], and p is in [0, 2π). Then forming the matrix, vector product $T|q> = (2\cos(t/2) + i\, e^{ip} \sin(t/2)\ \ e^{ip} \sin(t/2))'$. Use the one by two row vector to multiply the two by one column vector, thereby forming the inner product, resulting in the norm squared.

This gives $\|T|q>\|^2 = (2\cos(t/2) - i\, e^{-ip} \sin(t/2)\, e^{-ip}\sin(t/2)) \cdot (2\cos(t/2) + i\, e^{ip}\sin(t/2)\, e^{ip}\sin(t/2))' = 4\,(\cos(t/2))^2 + 2i\,(-e^{-ip}\sin(t/2)\cos(t/2) + e^{ip}\sin(t/2)\cos(t/2)) + (\sin(t/2))^2 + (\sin(t/2))^2 = 2 + 2\,(\cos(t/2))^2 + 4\sin(p)\sin(t/2)\cos(t/2) = 2 + 2(\cos(t/2))^2 + \sin(p)\sin(t)/2$. Accordingly, when $p = \pi/2$ and t is not too far from zero, the maximum of $\|T(v)\|^2$ is obtained, thus giving $\|T\|$ about equal to 2.25.#

Example 7.6:

The operator norm will again be found for the operator T: H → H where $H = C^2$; T is again given by the 2 x 2 complex matrix:

$$\begin{matrix} |2 & i| \\ |0 & 1| \end{matrix}.$$

This time, the SVD described in Section 5.8, in a modified form will be applied. An important use of the SVD with the complex scalar field is in finding the operational norm of a matrix operator. Here, $\|T\|$ is the square root of the largest eigenvalue of T*T. Calculating T*T, for the matrix above, yields:

$$\begin{matrix} |4 & 2i| \\ |-2i & 2| \end{matrix}.$$

The characteristic equation is $\lambda^2 - 6\,\lambda + 4 = 0$. The eigenvalues are about $\lambda_1 = 5.2$ and $\lambda_2 = 0.76$. So $\|T\| = (5.2)^{1/2}$, which is $(3 + 5^{1/2})^{1/2}$, and it is about 2.25 again.#

Example 7.7:

Consider the momentum-type differential operator D in $H = L^2\,[0, 2\pi]$. This is the set of all complex-valued square-integrable functions on $[0, 2\pi]$. D is defined on the set C^1 of continuously differentiable functions, which is dense in H since all polynomials are in this set. The sequence, e^{tni}, $n = 0, 1, 2, \ldots$, are in C^1 along with their derivatives with respect to t. These derivatives are $D(e^{tni}) = ni\, e^{tni}$. The values for the norms squared, in this case, are integrals of absolute value squared quantities. Therefore, using the inner product and conjugating the first entry in the integral whose limits are 0 to 2π gives $\|D(en^{tni})\|^2 = \int [-ni\, e^{-tni}\, ni\, e^{tni}\, dt]$, which equals $2\pi\, n^2$. Also, $\|e^{tni}\|^2 = \int [e^{-tni}\, e^{tni}\, dt] = 2\pi$ and $\|D(en^{tni})\| / \|en^{tni}\| = n$. This shows that the momentum operator is not bounded in H.#.

Bounded operators T and continuous operators are one and the same. In fact, bounded or continuous operators are Lipschitz continuous, which is even stronger than uniform continuity. To see this, let T be bounded, so that $\|T(v)\|$ is less than or equal to $M\,\|v\|$ for all v in the Hilbert space H. Letting $v = w - z$ gives the desired Lipschitz condition: $\|T(w - z)\|$ is less than or equal to $M\,\|w - z\|$. Going the other way, using the contrapositive, if T is not bounded, then it will be shown that T is not continuous. For every positive integer n, since T is assumed not bounded, there is a unit vector v_n in H with $\|T(v_n)\|$ greater than or equal to n. Taking the sequence $w_n = (v_n / n)$, which goes to zero as n goes to infinity, shows that T is not continuous at zero. This follows since $\|T(w_n)\| = \|T(v_n)\|/n$ is greater than or equal to one, as n goes to infinity.

More often then not, bounded operators do not commute. An important related concept is intertwining, a concept needed when spectral theory is described. Let P and S be bounded and have the same domain, with domain equal to their codomain. A bounded operator T, in the same space, is said to intertwine P and S, which means that for all v and w in H, it follows that: If P (v) = w, then S (T (v)) = T (w) = T (P (v)). When inverse operators exist for P and S, then T intertwines them also.

An additional property follows, namely that if S is in B (H1, H2) and T is in B (H2, H3), where H3 is also a Hilbert space, then T S = T (S (.)) is in B (H1, H3). Additionally, $\|T\ (S\ (v))\|_3$ is the norm involving H3. It is less than or equal to $\|T\|\ \|S\ (v)\|_2$, where the latter norm is in H2. This quantity is less than or equal to $\|T\|\ \|S\|\ \|v\|_1$, with the last norm in H1. Accordingly, $\|T\ S\|$ is less than or equal to $\|T\|\ \|S\|$.

The next section provides a pragmatic view for the definition of states in a Hilbert space, but first, a precise definition is the following: A state is a positive linear functional f, on a C* algebra, A, where f: A → C, and C is the complex field of scalars. Moreover, for all T in A, f (T*T) must be greater than or equal to zero, and most importantly, this functional operating on the C* identity element must equal one, f(I) = 1. In a later chapter, the most rigorous specifications such as these along with applications of states will be provided.

7.4 Pure tensors versus pure state operators

Recall that a state ρ on the Hilbert space H is a positive linear trace class map: H → H, such that Tr (ρ) = one. It is pure when there exists a vector v in H such that ρ (u) = [< v, u>/<v,v>] v. Pure states are denoted here by ρ_v.

An element z of Hilbert space H1 ⊗ H2 is said to be simple or called a pure tensor, which means that there exist vectors v in H1 and w in H2 such that z = v ⊗ w. This is wrongly called a pure state or sometimes a separable state in quantum computing. Fig. 7.1

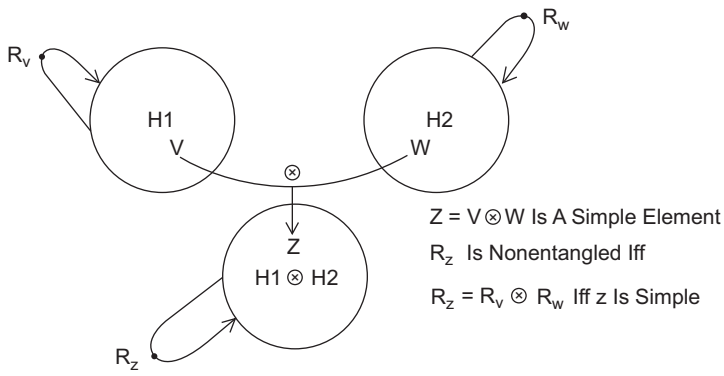


FIGURE 7.1 Simple elements in H1 ⊗ H2 and pure states.

R_z Is A Pure State

R_z = (< z, · >/ < z, z >) z

shows the formation of a simple element. The density operator is defined as a pure state. An operator ρ_z, which maps a Hilbert space H into itself, is said to be a density operator, which means that ρ_z is a linear trace class operator, $\rho_z\ (-) = (< - , z> / <z, z>)z$. In short, a trace class operator is T, such that the sum of the series of inner products of the form <ek, T ek> converges to the same value for any ON basis {ek} of H. The next section describes trace class operators in more detail.

A general element z of H1 ⊗ H2 involves some basis from H1 and H2, say

z = a11 e1 ⊗ f1 + a12 e1 ⊗ f2 + a21 e2 ⊗ f1 + a22 e2 ⊗ f2, where {e1, e2} and {f1, f2} are bases for H1 and H2, respectively. The tensor is called pure if the designated products of coefficients are equal, namely a11 a22 = a12 a21. Here, $|\ a11\ |^2 + |\ a12\ |^2 + |\ a21\ |^2 + |\ a22\ |^2 = 1$, and aij are in C.

Example 7.8:

The tensor $z = 2^{-1/2}\ (|0\ 1> + |1\ 1>) = 2^{-1/2}\ (|0 >\otimes |1> + |1> \otimes |1>)$. In this case, a11 = 0, a12 = $2^{-1/2}$, a21 = 0, and a22 = $2^{-1/2}$; therefore, this is a pure tensor.#

Example 7.9:

Next, for instance, one of the Bell states (Nelson, 2010), $z = 1/2^{1/2}\ (|0> \otimes |0> + |1> \otimes |1>)$, is a nonpure tensor or a nonsimple element of Hilbert space H1 ⊗ H2. This follows since a11 = $2^{-1/2}$, a12 = 0, a21 = 0, and a22 = $2^{-1/2}$.#

Additionally, any other basis utilized for describing this Bell state will again not satisfy the identity, a11 a22 = a12 a21. The next example illustrates the construction of a pure state and the map, ρ_z: H → H.

Example 7.10:

Let $H = C^2$, and consider the ket $|z> = 1/(2^{1/2}) \cdot (|0> + i \cdot |1>)$; the objective is to find the pure state map ρ_z corresponding to this simple element in H. Using the bra: $<z| = 1/(2^{1/2}) \cdot (<0| + - i \cdot <1|)$, then <z, z> = 1, and ρ_z is a pure state. Now, another simple element of H will be chosen, and we will see what the image of | u > is under ρ_z. Let $|u> = 1/(2^{1/2}) \cdot (|0> + - 1 \cdot |1>)$ since the inner product is <z, u> = (1 + i)/2; it follows that $\rho_z\ (u) = ((1 + i)/2) \cdot |z> = ((1 + i)/((2)^{3/2})) \cdot (|0> + i \cdot |1>) = (2^{1/2}\ e^{i\pi/4}/((2)^{3/2})) \cdot (|0> + i \cdot |1>) = 1/2 \cdot (|0> + i \cdot |1>)$.

The following are equational identities from Section 4.10. They are needed in describing entanglement and nonentanglement as explained below.

Equational identities for the tensor product of Hilbert space:

1) Linearity: c x (v1 ⊗ v2) = (c x v1) ⊗ v2 = v1 ⊗ (c x v2).
2) Distributive: (v1 + w1) ⊗ w2 = v1 ⊗ w2 + w1 ⊗ v2.
3) Distributive: v1 ⊗ (v2 + w2) = v1 ⊗ v2 + v1 ⊗ w2.
4) Inner Product: < v1 ⊗ v2, w1 ⊗ w2 > = <v1, w1> <v2, w2>.

A pure state ρ_z is said to be nonentangled in Hilbert space H1 ⊗ H2, which means there exist pure states ρ_v operating in H1 and ρ_w operating in H2, such that the tensor product: $\rho_v \otimes \rho_w = \rho_z$. A most important fact is that the pure state ρ_z is nonentangled iff for v in H1 and w in H2, it follows that z = v ⊗ w, that is, when z is a simple element of (or pure tensor in) H1 ⊗ H2. This can be seen by referring to the definition of tensor product of Hilbert spaces. Use the equational identity: (4) Inner product with v1 and w1 in H1, v2, and w2 in H2; then this identity is <v1 ⊗ v2, w1 ⊗ w2> = <v1, w1> <v2, w2>, so applying it to the pure state: ρ_z = (<z, ->/<z, z>) z = (<v ⊗ w, →/< v ⊗ w, v ⊗ w>) v ⊗ w = (<v, ->/<v, v>) (<w, ->/<w, w>) v ⊗ w. Then rearrange by using (1) Linearity in the definition of Hilbert space tensor: (<v, ->/<v, v>) v ⊗ (<w, −>/<w, w>) w = $\rho_v \otimes \rho_w$.

It was seen that a pure state ρ_z is said to be entangled in Hilbert space H1 ⊗ H2, which means there do not exist pure states ρ_v operating in H1 and ρ_w operating in H2, such that the tensor product $\rho_v \otimes \rho_w = \rho_z$ holds. A mixed state is entangled when and only when it cannot be written as a convex combination of pure states of the form $\rho_z = \rho_v \otimes \rho_w$. In the next example, we will show how tensor products of elements in a Hilbert space produce nonentangled states on a Hilbert space.

Example 7.11:

As an illustration of nonentangled pure state ρ_z in H1 ⊗ H2, consider ρ_u and ρ_v both pure states in H1 = H2 = C^2, respectively. Assume that $\rho_z = \rho_u \otimes \rho_v$ and that |u> = (1/(1.414)) • (|0> + |1>), and |v> = (0.8 x |0> −0.6 x |1>), with |u> in H1 and |v> in H2. Both elements are simple and the tensor product |z> = |u> ⊗ |v> = (1/(1.414)) • (0.8 • |0, 0> +0.8 • |1, 0> −0.6 • |0, 1> −0.6 • |1, 1>) is also simple in H1 ⊗ H2. This shows that the pure state ρ_z is nonentangled. Recall that the tensor is called pure iff a11 a22 = a12 a21; otherwise, it is entangled. Note that when this is represented as vectors in C^2, it follows that |z> = |u> ⊗ |v> = (1/(1.41)) • (0.8 • (1 0 0 0)′ +0.8 • (0 0 1 0)′ −0.6 • (0 1 0 0)′ −0.6 • (0 0 0 1)′) =

$$(1/1.41)\begin{matrix} |\ .8\ | \\ |-.6| \\ |\ .8\ | \\ |-.6| \end{matrix}.$$

The norm of | z > is 1.#

For the z in the Hilbert space H1 ⊗ H2, with normalized coefficients, $|a11|^2 + |a12|^2 + |a21|^2 + |a22|^2 = 1$, replacing ajk entries by real plus i times its imaginary part gives the equation of a seven sphere S^7. From a previous section, it was seen that a single qubit Hilbert space is S^3 and so the two qubit Hilbert space is S^7. Hopf fibrations describe these spheres in greater detail. Next, trace class operators are described in more generality. Subsequently, these operators are seen to be closely related to the Hilbert-Schmidt operators.

7.5 Trace class operators

Trace class operators generalize to infinite dimensions from the trace operation on square matrices. These operators are defined on Hilbert spaces, and the set of all trace class operators form a Banach space. They are closely related to the space of compact operators as well as the bounded operators. In fact, the dual space of trace class operators is the space of bounded operators. Additionally, the dual space of compact operators is the space of trace class operators (Argerami, 2014).

For a linear operator T, on a Hilbert space H, the trace of T is the sum of the series of inner products of the form <ek, T ek>, where {ek} is an ON basis of H. For an infinite-dimensional Hilbert space, the series must converge. As usual, it is the limit of the sequence of partial sums. In this case, it is denoted by Tr (T), and the result must be the same regardless of the basis. Again, to be a trace class, the trace must be finite no matter what basis is employed, and all the limits of the defining sum must be equal.

Example 7.12:

Let H be a finite, n-dimensional Hilbert space. For any n by n matrix M, the trace defined earlier agrees with the usual trace operation on matrices. No matter which ON basis is employed, it must follow that Tr (T) always yields Trace (M). For instance, use the carrier set $H = C^2$. Let the basis set be $\{i|0>, -1\ |1>\} = \{(i\ 0)', (0\ -1)'\}$; then using these 2 by 1 vectors along with M =

$$\begin{matrix} |4-i & -2| \\ |3 & -2| \end{matrix}$$

gives Tr (M) = < (i 0)′, M (i 0)′> + < (0 −1)′, M (0 −1)′ >. So: Tr (M) = < (i 0)′, (4i + 1 3i)′> + < (0 −1)′, (2 2)′>. Now recall that the inner product is conjugate linear and so the left operand should be conjugated. Therefore, Tr (M) = 4 − i − 2 = 2 − i. By inspection, the sum of the diagonal elements of M is the trace of M. The result agrees with 2 − i. #

A generalization of this example is that any bounded linear operator with a finite-dimensional range, that is, whose codomain is of finite cardinality, is of trace class. However, in general, infinite-dimensional situations can be different. The next example is for an infinite-dimensional Hilbert space, H. The operator in this example is not of trace class even though it gives a finite trace for a single ON basis, but not for all ON bases for H.

Example 7.13:

Let B = {e0, e1, e2, ….} be an ON basis for H. It will be shown that an operator T has trace equal to zero with respect to the basis B, but it has infinite value with respect to basis F. Let $F = \{(1/\ 2^{1/2})\ (e1 - e0), (e2 - e1), (e3 - e2), \ldots.\}$; it is also an ON basis. This can be seen by forming the inner product: <fi, fj> = (1/ 2) <ei + 1 − ei, ej + 1 − ej> = 1 when

i = j and 0 otherwise. Next, let T: H → H, where T sends the basis B into itself by transposing adjacent elements. It exchanges even entries for odd ones next to each other and vice versa, that is, T(e2n) = e2n + 1, T (e2n + 1) = e2n, n = 0, 1, 3, …. Note that the infinite sum $\sum_i$ <(ei), T ei> = 0. But the infinite sum employing basis F is $\sum_n$ <((1/ 2)$^{1/2}$)(e2n + 1 − e2n), (1/ 2)$^{1/2}$T(e2n + 1 − e2n)> = the infinite sum (1/ 2) $\sum_n$ <e2n + 1 − e2n, e2n - e2n + 1> = infinite sum 1/ 2 $\sum_n$ − ($\|e2n\|^2$ + $\|e2n + 1\|^2$) = − ∞.#

Trace class operators are similar to Hilbert-Schmidt operators defined in the next section.

7.6 Hilbert-Schmidt operators

This section begins with one of the most important inequalities in Hilbert spaces. It is the Cauchy-Bunyakovsky-Schwarz inequality, and throughout the document, it is abbreviated as CBS inequality. This inequality holds for any vectors v and w in a Hilbert space. However, a similar inequality exists for positive sesquilinear forms in any vector space.

The CBS inequality is |<v, w>|, which is always less than or equal to the product $\|v\|$ $\|w\|$. This result follows by considering the nonnegative inner product: N = < v + a t w, v + a t w>. Here, a is in C and t is in R. To see that this inequality holds, expand the inner product yielding N = $\|v\|^2$ + a* t <w, v> + a t <v, w> + |a|^2 t^2 $\|w\|^2$. Next, by expressing the complex constant a, as a = < w, v>, then if this value is substituted in the aforementioned expression for N, it follows that N = $\|v\|^2$ + 2 t | <v, w>|^2 + |<v, w>|^2 t^2 $\|w\|^2$. Observe that N is a quadratic polynomial; it is a parabola in t. Because it is nonnegative, it touches the x-axis at most once. Accordingly, its discriminant is negative or zero. So, the discriminant, 4 | <v, w> |^4−4 $\|v\|^2$ | <v, w>|^2 $\|w\|^2$, is negative or zero. Notice that if <v, w> = 0, there is nothing to do. Otherwise, factoring out |<v, w>|^2 shows that the CBS inequality holds. Moreover, from the initial representation of N, the equality sign holds if and only if v and w are linearly dependent. An application of the CBS inequality follows.

Example 7.14:

Let v and w be vectors in the Hilbert space H. Using the operator norm type of notation, it will be shown that $\|v\|$ = sup |<v, w>| where $\|w\|$ = 1. If v is zero, the result follows immediately. Otherwise, let b = sup |<v, w>| where $\|w\|$ = 1. By the CBS inequality, it is seen that |<v, w>| is less than or equal to $\|v\|$, and so b is less than or equal to $\|v\|$. Next, it will be shown that b is greater than or equal to $\|v\|$. Letting w = av, then |<av, v> | = | a | $\|v\|^2$. Since b is found utilizing the supremum of |<v, w>|, the result is that b is greater than or equal to |a| $\|v\|^2$. Finally, since w = av, |a| = $\|w\|$/ $\|v\|$ = 1/$\|v\|$, and $\|v\|$ is not zero, it is found that b is greater than or equal to $\|v\|$, and so b = $\|v\|$.#

The following operators are similar to the trace class operators mentioned previously. Here, the Hilbert-Schmidt bounded operators T are defined in the Hilbert space H when n is

any integer and vn is an ON basis. In this case, $A = \sum_{n=-\infty}^{\infty} \|Tvn\|^2$ converges. Let wk and vk be the same ON set, and notice that the Hilbert-Schmidt norm for bounded T is given by $\|T\| = \left[\sum_{n=-\infty}^{\infty} \|Tvn\|^2\right]^{1/2}$. This norm was used for matrix norms earlier and is also called the Frobenius norm. See Section 5.1. When T is only a linear operator, the defining sum could be positive infinity. Including the point at infinity with $[0, \infty)$ results in a pointing set.

Additionally, the identity holds, $\|T\| = \|T^*\|$. This can be seen by first expressing the norm $\|T\ v\|$ for the operator T involving the inner product using an ON set of vectors wn, in H. Accordingly, this results in $\|T\ v\|^2$ equaling the sum over all the integers of $\sum_n |< wn, Tv >|^2$. In the following, the sum will always be from $-\infty$ to ∞, unless otherwise specified, $\|T\|^2 = \sum_n \sum_k |< vk, T\ wn>|^2 = \sum_n \sum_k |< T^*\ vk, wn >|^2 = \sum_k \sum_n |<T^*\ vk, wn >|^2. = \sum_k \| T^*vk \|^2 = \|T^*\|^2$. Thus, it follows that $\|T\| = \|T^*\|$.

Notice also that from the CBS inequality, it can be seen that $\|T\ v\ \|^2$ is less than or equal to $\|T\|^2\ \|v\|^2 = \|T^*\|^2\ \|v\|^2$.

Again, employing the CBS inequality will show that a bounded Hilbert-Schmidt operator is also compact. Let Tn be a sequence of finite rank operators. As such, each Tvn is represented for any v in H as a finite sum: k = 1, 2, 3, . . ., N of <wk, Tn v> wk, with wk, as before being an ON basis in H. Form the quantity $\|(T - TN)v\|^2$, notice that from it is the truncated infinite sum k = N + 1, N + 2, . . ., of $|< wk, T\ v>|^2$. Then, using the CBS inequality, $\|(T\ - TN)\ v\|^2$ is less than or equal to the tail of the series of $\|T^*\ wk\|^2\ \|v\|^2$, but for k = N + 1, N + 2, . . ., the sum of $\|T^*\ wk\|^2$ goes to zero.

The set of all Hilbert-Schmidt operators, S and T, form a vector space, a subspace of the bounded operators. Moreover, the following inner product can be used and is substantiated using CBS inequality: $<S, T> = \sum_n <S\ vn, T\ vn>$, where vn is an ON basis in H. Additionally, the vector space of all Hilbert-Schmidt operators forms a Hilbert space (Conway, 1990).

7.7 Compact operators

The notion of a relatively compact set S, in a complex Hilbert space H, must be given. It is such that for any sequence vn with values in S, there must exist a subsequence of vn, which converges to a value in S. A linear operator T, T: H1 → H2 with H1 and H2 Hilbert spaces, is said to be a compact operator when T (B (0, 1)) is relatively compact in H2. Here, B (0, 1) is an open ball centered at 0 and radius 1 in H1. This result could be defined in a more general setting; for instance, the result holds for a Banach space that was introduced in a previous chapter. It was previously seen that the MSA provides a platform to globally abstract properties of the Banach spaces, Banach algebras, and C* algebras. These algebras form a foundation for bounded operators used in quantum disciplines. A simple example of a closed set S, which is not relatively compact in a Hilbert space, is provided next.

Example 7.15:

Consider the closed and bounded unit ball, B_C (0, 1), in the Hilbert space H. Let e_1, e_2, ... be an ON sequence for H, since $\|e_i\| = 1$; this is a sequence of points on the boundary of this ball. However, the inner product using distinct ON vectors gives $< e_i - e_k, e_i - e_k > = < e_i, e_i > + < e_k, e_k > = 2$. So these points are $2^{1/2}$ apart, and therefore, there is no convergent subsequence. As such B_C (0, 1) is not compact.#

In a bounded linear transformation, T is of finite rank whenever the range space is of finite dimensions. For this type of operation, T (B) is a bounded subset in C^n. Its closure is closed and bounded, and therefore it is compact. It follows that finite rank operators are compact operators. Additional results are provided involving compact operators in the forthcoming chapters. In particular, the spectral theorems involving these operators are easiest to prove.

References

Argerami, 2014. Stack Exchange.
Conway, J., 1990. 978-0-387-97245-9 A Course in Functional Analysis. Springer-Verlag.
Gustafson, K., 1970. The Toeplitze-Hausdorff Theorem for Linear Operators. American Math Society.
Nelson, M., 2010. ISBN9781139495486 Quantum Computation and Quantum Information. Cambridge U. Press.
Rudin, W., 1991. second ed. Functional Analysis, International Series in Pure and Applied Mathematics, vol. 8. McFraw-Hill.

CHAPTER

8

Spaces and algebras for quantum operators

8.1 Banach and Hilbert space rank, boundedness, and Schauder bases

This section begins by again showing that partial operators on a Banach space, for instance, the momentum operation in quantum studies, are unbounded. In Chapter 7, Example 7.7, it was seen that the differential operator is unbounded in the Hilbert space, L^2 [0, 2π]. Now it will be shown that it is also unbounded in the Banach space of continuous functions.

Example 8.1:

Consider the space: B = C([0, 1]) with norm of g, $\|g\| = \max |g(t)|$ for t in [0, 1]. A subspace of B will be employed for a partial operator in B. So, let the differential operator be T = d/dt. The differential operator is not defined in this Banach space; instead, it is defined in the linear subspace of continuously differentiable functions, C^1 ([0, 1]), a subset of B. However, T is a closed operator since if $f_n \to f$ and $f'_n \to g$, then $Tf = g = f'$. This follows assuming that the convergence above is both uniform convergence on [0, 1] and f is in C^1. To see this, write $f_n(t) = f_n(0) + \int_0^t f_n'(x)\,dx$ and take the limit as n goes to infinity. Also interchange the integral with the limits. This follows since uniform continuity prevails. The result is $f(t) = f(0) + \int_0^t g(x)\,dx$. Taking the derivative of the last equation gives $f'(t) = g(t)$, which shows that the operator d/dt is a closed operator.

It should be pointed out that the assumption of uniform convergence is crucial. See what happens when $f'_n \to g$, only point-wise. Consider $f_n(t) = t/(1 + n\,t^2) \to f = 0$, for all t in [0, 1], and so f is in C^1[0, 1] with $Tf = f' = 0$. On the other hand, $T f_n = f'_n = (1 - n\,t^2)/(1 + n\,t^2)^2 \to g$, where $g(t) = 0 = T f(t)$, for all t in (0, 1], but $T f_n(0) = f'_n(0) = 1$ and is not equal to T f (0).

Next, it will be shown that T is not bounded.

The operator T is linear, but is not a continuous operator since it is not bounded. This can be seen by letting $f(x) = x^n$. Then the norm of the derivative $\|T\,x^n\| = \|(x^n)'\| = n\|x^{n-1}\|$. Taking the maximum for x in [0, 1] shows that $\|T\|$ is of order n, as n goes to infinity.#

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00005-9

A Schauder basis in a Banach or Hilbert space V is a sequence of vectors: v1, v2,. . ., such that for any v in V, there are unique scalars a1, a2,. . ., which depend on v, and where v = a0 • v0+a1 • v1+a2 • v2 + . . ., such that this series converges in norm.

A Banach space can become a Hilbert space iff the parallelogram identity holds, that is, $\|u+v\|^2+\|u-v\|^2 = \|u\|^2+\|v\|^2$. The parallelogram law alone is not a sufficient condition to obtain an inner product; consequently, more is needed. Specifically, the polarization identity provides the extra criteria and is defined next. It enables an inner product to be constructed such that the resulting space is an inner product space. The polarization identity for the complex field is $<u, v> = 1/4\ (\|v+u\|^2-\|v-u\|^2+i \cdot \|v+i \cdot u\|^2-i \cdot \|v-i \cdot u\|^2)$. For the real field, only the first two entries appear, that is, $<u, v> = 1/4\ (\|v+u\|^2-\|v-u\|^2)$.

An example of a Banach space that is not a Hilbert space is given next. This example also uses the fundamental position operator in quantum settings.

Example 8.2:

Consider the space B = C([0, 1]) with norm of g in B given by $\|g\| = \max\ |g(x)|$ for x in [0, 1], as used in Example 8.1. This time, we show that this norm vector space is not an inner product space. Therefore, it is also an illustration of a Banach space, which cannot become a Hilbert space. The principle tool employed is the parallelogram law. To begin, let f also be a vector in B. The parallelogram law does not hold because the identity, $\|f+g\|^2+\|f-g\|^2 = \|f\|^2+\|g\|^2$, does not hold. By using f = g = x, then $\|g\| = \max\ |g| = \|f\| = \max |f|$ for x in [0, 1] is one, but $\|f+g\|^2 = 4$.#

An operator T from a Banach space B1 into a Banach space B2 is said to be of finite rank means that the image of T is a finite-dimensional subspace of B2. Any operator in B (B1, B2) means that T is a linear bounded operator from B1 to B2. If it is also of finite rank, then T is compact. Here, the image of T is in the open ball of radius $\|T\|$ centered at 0 in B2. More generally, the set of all compact operators from B1 into B2 is itself a vector space subspace of B (B1, B2). Additionally, if the sequence T0, T1, T2,. . . in B (B1, B2) of finite rank operators converges to T, then T is a compact operator (Conway, 1990).

In particular, when T in B (B1, B2) is of rank one, then there exists a vector v of norm one in B1 such that a unique functional f (w) exists where T w = f (w) v for all w in B1. The linearity of T shows that f is linear and that $|f(w)| = \|T\ w\|$ is less than or equal to $\|T\|\ \|w\|$, showing that f is bounded. If this is also a Hilbert space H, then by RRT, that is, by the Riesz representation theorem, this bounded linear functional is also an inner product; thus, f (w) = <u, w> for some u in H and so T w = <u, w> v. In general, for a finite operator T of rank n, the ON set {v1, v2,. . ., vn} can be used, and the corresponding functionals to obtain $T\ w = \sum_{k=1}^{n} <uk, w> vk$. A very interesting example follows. It shows that in a Banach space there exists unbounded operators of finite rank (Lacey, 1973).

Example 8.3:

Let B1 be an infinite-dimensional Banach space over C. An example of an unbounded operator T, which has finite rank, will be given. Say that M is a Hamel basis of B1. This basis always exists by application of the axiom of choice, and it is uncountable. Let L be a countably infinite subset $\{e_n, n = 1, 2,. . .\}$ of M. Let T: L→C, where $T(e_n) = n \cdot e_n$, and

T(v) = 0 for all v in M−L. The dimension of (M−vL) is at least equal to the continuum. T has range dimension 1 and therefore is of finite rank but is not a bounded operator.#

8.2 Commutative and noncommutative Banach algebras

Although all Banach algebras have the same global MSA structure, a great difference appears in the carrier sets for these algebras. Among the largest distinguishing factors is whether they are commutative or not. This is illustrated by numerous examples.

Example 8.4:

Let the carrier set for VECTOR be all n by n complex matrices, A. It is a noncommutative Banach algebra over the complex field. Let NORM utilize this carrier set along with the corresponding carrier set consisting of complex numbers, as well as all usual complex matrix operations. Here, let the operation for NORM be the square root of the sum of all absolute squared entries of A. This is the Frobenious or Hilbert-Schmidt norm. See also Section 7.6. The Frobenious norm satisfies the four constraining equations for a norm. To see this, first replace:

MATRIX by A, B
SCALAR by a
C-MULT by x
V-ONE by: I
BINE by ·
NORM by $\| \ \|$

1) Positive Definite: $\|A\| > 0$, and $\|A\| = 0$, iff $A = 0$.
2) Homogeneous: $\|a\ A\| = |a|\,\|A\|$
3) Triangle Inequality: $\|A+B\|$ is less than or equal to $\|A\|+\|B\|$
4) Triangle Product Inequality: $\|A \cdot B\|$ is less than or equal to $\|A\| \ \|B\|$.

When (4) holds, the norm is said to be submultiplicative.

To see that (4) holds, using the Frobenius norm, say that $D = A \cdot B$ in C^n. Then a typical element in D is $d_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$. Also, $|d_{ij}|^2 = |\sum_{k=1}^{n} a_{ik} \cdot b_{kj}|^2$ by the CBS inequality; the result is that $|d_{ij}|^2$ is less than or equal to $\left[\left(\sum_{k=1}^{n} |a_{ik}|^2\right) \cdot \left(\sum_{k=1}^{n} |b_{kj}|^2\right)\right]$. Therefore, $\|D\|^2 = \|A \cdot B\|^2$ is less than or equal to $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{m=1}^{n} \sum_{k=1}^{n} |a_{im}|^2 \cdot |b_{kj}|^2) = \left(\sum_{i=1}^{n} \sum_{m=1}^{n} |a_{im}|^2\right) \cdot \left(\sum_{j=1}^{n} \sum_{k=1}^{n} |b_{kj}|^2\right) = \|A\|^2 \cdot \|B\|^2$.#

Example 8.5:

An important result is that for T in B(H) and for v and u unit vectors in H, then $\|T\| = \sup\ |<Tu, v>|$. The result follows from a previous example, that is, Example 7.14 will be employed. There, it is shown that $\|w\| = \sup\ |<w, v>|$ where $\|v\| = 1$.

So employ the referenced example by noticing that Tu is a vector in H. Accordingly, let w = Tu; then, from the previously mentioned example, $\|Tu\| = \sup|\langle Tu, v\rangle|$, because $\|v\| = 1$. Use the operator norm $\|T\| = \sup|Tu|$, when $\|u\| = 1$. Consequently, by backtracking substitution, $\|T\| = \sup|\sup|\langle Tu, v\rangle|| = \sup|\langle Tu, v\rangle|$, as desired.#

Example 8.6:

A simple example of a commutative Banach algebra involves a carrier set of all bounded functions in a nonempty subset S of the complex plane. It is denoted by A. For any functions f and g in A, it follows that the point-wise operations of addition, (f+g) (z) = f (z)+g (z), are closed. The same is true for scalar multiplication, where the scalar a is in C, a (f (z)) = af (z). Moreover, all the equational identities hold for a vector space structure. Additionally, it is a unital associative algebra, that is, f • g is in A. This follows since (f g) (z) = f (z) g(z), the V-ONE is 1, and |1| = 1. Finally, it is a Banach algebra using the norm $\|f\| = \sup|f(z)|$ for z in S and observing that the triangle product inequality trivially holds. In fact, it is an equality, |f • g| = |f| • |g|. Additionally as in a C* algebra, |f • g*| = |f| • |g*|.#

Example 8.7:

The position P, and momentum M, operators do not commute. The following will only be a formal presentation of the results. See the reference mentioned later. Noncommuting of P and M can be seen using the commutator operation that is [P, M| = P M −M P], which is basic to Lie algebras. To commute means that this Lie bracket is zero. However, it equals i multiplied by Plank's constant h times the identity I, operation. This result follows by taking any continuously differentiable f, and xf in the Hilbert space of absolute square integrable functions, L2. Then employing the momentum operator P and multiplication operator M. Here, P is defined using Pf = −ihd(f(x))/dx, and the multiplication operator is Mf = xf(x). So by the product rule for differentiating, PMf(x) = −ihd(x f(x))/dx = −ih(f(x))+x d(f(x))/dx = −hif+MPf(x), and therefore the identity [P, M] = −ihI or [M, P] = ihI holds. This identity is often called the canonical commutation relations (CCRs) (McCoy, 1929).#

This CCR identity can be used in showing that either P or M is an unbounded operator. Standard induction techniques are employed in showing unboundedness. Begin with $[M^n, P] = M^n P - P M^n = ih\, n\, M^{n-1}$. This is true for n = 1, [M, P] = ihI holds, and if it is assumed true for n, then by induction it must be shown true for n + 1. That is, it must be shown that the following expression holds: $M^{n+1}P - P M^{n+1} = ih(n+1)M^n$. The left hand side can be written as follows: $M^n (M P - P M) + (M^n P - P M^n) M = M^n ih + ih\, n\, M^{n-1} M =$ the right hand side, $ih(n+1) M^n$. Now bounding the right hand side: for $[M^n, P]$ gives the bound: $n \|(M^{n-1})\|$ is less than or equal to $\|M^n\| \|P\| + \|P\| \|M^n\|$, and so factoring out $\|M^{n-1}\|$ gives n is less than or equal to $2\|M\|\|P\|$. Therefore at least one M or P is not bounded.

An interesting example of a real Banach algebra is given next involving an important subset of continuous real-valued functions. They are the almost periodic functions, a generalization of periodic functions. Among the most famous is the Bohr-type uniformly almost periodic continuous function f. This means that for every a greater than zero, there exists a relatively dense subset Da of R, such that $\sup|f(x+t)-f(x)| < a$, for all t in Da. The value t is called an a translation number. Intuitively, this means that even though there

may not exist a value t such that f(x+t) = f(x) for all real values x, there may be numbers that make this relation an approximate equality. The concept is best illustrated by the following example (Gelbaum and Olmsted, 2003).

Example 8.8:

For f: R→R, consider f (x) = sin (2πx)+sin ($2^{1/2}$ 2π x); this function is not periodic. However, for any ϵ> 0, it can be shown that there are an infinite number of integer values t, such that $2^{1/2}$ • t, differs from another integer by less than or equal to ϵ/(2π). Moreover, the difference between two consecutive such integers is bounded, by δ. Using one such integer value, t, then f (x+t) = sin (2πx+2π t)+sin ($2^{1/2}$ (2π x+2π t)) = sin (2πx)+sin ($2^{1/2}$ 2π x+δϵ) = f (x)+δ′ϵ, where δ′ is a bounded quantity. The basic idea is that f comes close to being periodic, but it is not periodic.#

Almost periodic functions are well behaved. They are uniformly bounded and uniformly continuous. Additionally, if f_n is a sequence of almost periodic functions that converges uniformly to f, then f is also almost periodic. The space of all almost periodic functions is a closed subspace of all bounded and continuous functions. Under the sup norm, it is a Banach space. Using point-wise multiplication of functions shows that it is a Banach algebra. An instance of a space of almost periodic functions is all the trigonometric polynomials. These are finite sums over k an integer, involving linear combinations of sinusoids, (b_k cos(d_k x)+ c_k sin(d_k x)), with b_k, c_k, and d_k in R.

The space of almost periodic functions forms a pre-Hilbert space using the inner product: <f, g> = limit as T goes to infinity of [1/2 T $\int_{-T}^{T}$ f(x)*g(x) dx].

This space is completed to the class B^2 often called the class of Besicovitch almost periodic functions (Besicovitch, 1954).

8.3 Subgroup in a Banach algebra

An important subset of a Banach algebra A is the group G of invertible elements from A. Here, from the MSA perspective, there is a single sort, SUBG. Concentrating on the algebraic properties of A, there are three signature sets each containing a single operational name. The arity sequence therefore is (1, 1, 1). The operational names in each signature set agree with those from the superset, namely the Banach algebra. Additionally, however, there is the new operator name INV. It is given in a suggestive fashion. There are no partial operators in this case. The arities with corresponding operational names are the following:

Arity 2:BINE:SUBG x SUBG→SUBG

Arity 1:INV:SUBG→SUBG

Arity 0:V − ONE

Three equational identities must hold because this is a similar algebra to the additive group structure within a vector space. Before these identities are given again, several

symbols are assigned representing the specified sort. Additionally, suggestive symbols are provided for the operational names. These are given for

SUBG by u, v, w
BINE by •
INV by /
V-ONE by I

The equational identities are as follows:

1) Associative for multiplication: $(u \cdot (v \cdot w)) = ((u \cdot v) \cdot w)$.
2) One law: $I \cdot v = v \cdot I = v$, I is V-ONE.
3) Inverse law: using $1/v$, then $v \cdot 1/v = 1/v \cdot v = I$.

In the Fig. 8.1, the names of operations in a subgroup of a Banach algebra are illustrated. An important topological property of the group G of invertible elements from the Banach algebra A is that G is an open set within A. This means that for every point within G, this point is an interior point. There exists a sphere or ball of radius ϵ about that point such that the sphere lies wholly within G.

The next few examples illustrate the interplay between the (sub) group G, within the Banach Algebra A, and A itself. The next example is particularly important in spectral analysis. It involves the identity function I, as well as many of the inherited properties that the group G obtains from the Banach algebra A. For instance, the norm and convergence properties from A carry throughout the subgroup G.

Example 8.9:

Let v be in the Banach algebra and A be such that $\|v\|$ is less than one, and set $r = I - v$. Then r has an inverse. This result follows from a power series argument. The series involved is often called the Neumann series. Let $w_n = v + v^2 + \ldots + v^n$; then, the sequence of partial sums is a Cauchy sequence, and therefore, it converges to some vector in A. Note that $(I - v)\ w_n = w_n\ (I - v) = (I - v^{n+1})$, and this quantity converges to I as n goes to infinity. Likewise, as n goes to infinity, the sequence of partial sums converges to say w. Additionally, $(I - v)\ w = w\ (I - v) = I$, that is, $r\ w = w\ r = I$, by continuity of multiplication. Also, $w = I/r = 1/(1 - v) \sum_{n=0}^{\infty} v^n$ will be the standard notation for the inverse when it exists.#

An instance of criteria for vectors within A to have an inverse is illustrated in the following example.

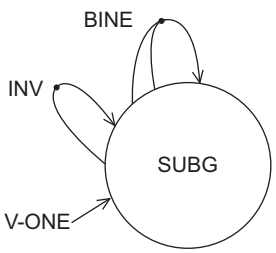


FIGURE 8.1 Polyadic graph for subgroup in Banach algebra.

Example 8.10:

Let v be in G and w be in the Banach algebra A. Assume that $\|v-w\|$ is less than $1/\|v^{-1}\|$; then w is also in G, that is, it is invertible. To see this, consider the vector z in A, where $z = I - v^{-1} w$. Then, $\|z\| = \|I - v^{-1} w\| = \|v^{-1}(v-w)\|$ is less than or equal to $\|v^{-1}\| \|(v-w)\|$, which is less than $\|v^{-1}\|/\|v^{-1}\| = 1$. Because, $\|z^n\|$ is less than or equal to $\|z\|^n$ for nonnegative integers, it will be shown that the summation $\sum_{n=0}^{\infty} z^n$ converges to say y in A. That is, the sequence of partial sums $\sum_{n=0}^{N} z^n = (I - z^{N+1})/(I - z) = \left(I - (I - v^{-1}w)^{N+1}\right)/(v^{-1}w)$ converges to $y = 1/(1-z)$. Accordingly, $y = 1/(v^{-1} w)$.

To validate the convergence, absolute convergence will be illustrated. Note that $\|I - y\|$ is less than or equal to $\sum_{n=1}^{\infty} ||z||^n = \sum_{n=1}^{\infty} ||v^{-1}(v-w)||^n$, which is less than or equal to $\sum_{n=1}^{\infty} \left(\|v^{-1}\| \|v-w\|\right)^n = 1/[1 - (\|v^{-1}\| \|v-w\|)] - I = \|v^{-1}\| \|v-w\|)/[1 - \|v^{-1}\| \|v-w\|)]$.

Multiplying, $y(I-z) = (1-z)\, y$, and then substituting in for z, $z = I - v^{-1}w$, gives $y\, v^{-1}w = v^{-1}\, w\, y = I$. So, $y(I-z) = (1-z)\, y = I$; this shows that y and $(1-z)$ are invertible. Moreover, it follows that $v^{-1}\, w$ is invertible, and $v^{-1}\, w\, y = y\, v^{-1}\, w = I$. Thus, $w = v\, y^{-1}$ and $w^{-1} = y\, v^{-1}$, so w is in G.#

A final fact for the Banach algebra A is that if every nonzero element in A is invertible then A is isometrically isomorphic to C. The only complex Banach algebra that is also a division algebra is the complex field. This is the Gelfand-Mazur theorem (Bonsall and Duncan, 1973).

8.4 Bounded operators on a Hilbert space

Consider the set B(H) of all complex-valued bounded operators mapping the Hilbert space H into itself. The space of all these operators B(H) = B (H, H) is a C* algebra. It is illustrated step by step:

A) It is a vector space;
B) It is a unital, associative algebra;
C) It is a Banach space;
D) B (H, H) is a Banach algebra;
E) It is a Banach* algebra;
F) Finally, it is a C* algebra.

Here, T has the operator norm: $\|T\| = \sup\{\|T\, w\|/\|w\| \mid w \text{ in } H, w \text{ nonzero}\}$. It was seen that $\|T\| = \sup_{\|w\|=1} \|T\, w\|$ and that the involution is the usual adjoint, $<T^*v, w> = <v, T\, w>$. This is the set of all endomorphisms B(H) on H.

Fig. 3.5 in Chapter 3 illustrates all of the eight operational names in a C* algebra. These are actually the closure operations for B(H, H): V-ADD, +; S-MULT, •; V-MINUS, -; BINE, •; NORM, $\|\ \|$; ADJ, *; V-ZERO, 0; V-ONE, 1. Recall that BINE in this algebra is function composition. For S and T in B(H, H), BINE(T, S) = T • S = T(S (.)), so for v in H, BINE(T, S)(v) = T(S(v)). All the operations mentioned earlier are continuous. The continuity for the NORM is proved in Section 3.6, and the continuity of BINE is proved in

Section 3.7. That the operations in the referenced figure hold true, that is, the closure requirements are met, is a result of the Lipschitz condition: $\|T(v)\|$ is less than or equal to $M\|v\|$. Recall that $<T(v), T(v)> = \|T(v)\|^2$. The infimum of all such values of M is called the operator norm of T and is denoted by $\|T\|$. For instance, the V-ADD and the S-MULT operations are closed, as a result of the triangle inequality. That is, for a in C and v and w in H, notice that $\|a\,v+w\|$ is less than or equal to $|a|\|v\|+\|w\|$. Thus, V-ADD[S-MULT(a; v), w] is in B(H, H).

The following list of 23 equational identities hold for B(H, H). This listing presents the required laws for the algebraic structures (A) through (F). These are the equational identities that must hold true and are detailed below. In this list, let R, S, and T be operators in B(H, H), and also let a and b be scalars in C. After the listing of all the equational constraints, several proofs are provided, which were not proven previously.

A) The equational identities that hold for B (H, H) since it is a vector space are as follows:
- **1)** Associative for vector addition: $(S + (T + R)) = ((S + T) + R)$.
- **2)** Zero vector law: $0 + T = T + 0 = T$.
- **3)** Minus vector law: $T-T = -T + T = 0$.
- **4)** Commutative vector law for addition: $S + T = T + S$.
- **5)** One law: $1 \cdot T = T \cdot 1 = T$.
- **6)** Distributive law: $a \cdot (S+T) = a \cdot S+a \cdot T$.
- **7)** Distributive law: $(a+b) \cdot T = a \cdot T+b \cdot T$.
- **8)** Associative law: $(a \cdot b) \cdot T = a \cdot (b \cdot T)$.

B) The equational identities that hold for B (H, H) because it is an associative unital algebra are as follows:
- **1)** Distributive law: $(S+T) \cdot R = S \cdot R+T \cdot R$.
- **2)** Distributive law: $R \cdot (S+T) = R \cdot S+R \cdot T$.
- **3)** Multiplicative homogeneity: $ab\,(S \cdot T) = (a\,S) \cdot (b\,T)$.
- **4)** Associative law: $((R \cdot S) \cdot T) = (R \cdot (S \cdot T))$.
- **5)** V-ONE law: $I \cdot T = T \cdot I = T$.

C) B (H, H) satisfies the following conditions for a NORM since it is a Banach space:
- **1)** Positive definite: $\|T\| > 0$, and $\|T\| = 0$, iff $T = 0$.
- **2)** Homogeneous: $\|a\,T\| = |a|\|u\|$.
- **3)** Triangle inequality: $\|S+T\|$ is less than or equal to $\|S\|+\|T\|$.

D) The additional condition it satisfies because it is a Banach Algebra is as follows:
- **4)** Triangle product inequality: $\|S \cdot T\|$ is less than or equal to $\|S\|\,\|T\|$.

E) Additionally B (H, H) is a Banach* algebra since the following laws are obeyed:
- **1)** Involuntary: $(T^*)^* = T$.
- **2)** Additive: $(S+T)^* = S^*+T^*$.
- **3)** Conjugate linear: $(a\,T)^* = \bar{a}\,T^*$. Here, $\bar{a}$ denotes the conjugate of a.
- **4)** Transpose: $(S \cdot T)^* = T^* \cdot S^*$.
- **5)** Isometry: $\|T^*\| = \|T\|$.

F) B (H, H) is C* algebra since it is a Banach* algebra also satisfying:
- **6)** C* Identity: $\|T \cdot T^*\| = \|T\|^2$.

Proofs of some these equational identities:

C 3) The triangle inequality is shown in Section 7.3.
D 4) The triangle product inequality $\|S \cdot T\|$ is less than or equal to $\|S\| \|T\|$ followed by the function composition $\|S(T)(v)\|$. Since $\|T(v)\|$ is less than or equal to $\|T\| \|v\|$, for all v in H. Using $w = T(v)$, then $\|S(w)\|$ is less than or equal to $\|S\| \|w\|$, this holds for all w in H.
E 5) To see that the Banach* algebra isometry identity holds, use the result from Example 8.5 in Section 8.2. From this example, note that the operator norm can be written as $\|T\| = \sup| < Tv, w > |$ for v and w in H each of unit norm. So, $\|T^*\| = \sup| < T^*v, w > | = \sup| < v, Tw > | = \sup| < Tw, v >^* | = \sup| < Tw, v > | = \|T\|$.
F 6) To prove the C* identity, again use the result from previous examples. Begin with $\|T^* T\|$ is less than or equal to $\|T^*\| \|T\| = \|T\|^2$. Next the inequality will be reversed. As in equation, E 5). For v and w in H, each of unit norm $\|T^* T\| = \sup| < T^*T w, v > | = \sup| < T w, Tv > |$ is greater than or equal to $\sup| < T v, Tv > | = \|T\|^2$. Thus, $\|T^* T\| = \|T\|^2$.

8.5 Invertible operator algebra criteria on a Hilbert space

Consider bounded operators T in A such that T: H→H, that is, T is in B (H, H). The concept of inverse is extremely important in an operator algebra A, because the spectrum of T depends on the inverse existing or not. In particular, this spectrum is denoted by spT and is defined to be the set of all complex numbers c, such that $(c \cdot I - T)$ is not invertible; here I is the identity element in A. To be clear, an operator S in A is invertible iff its image or range {Sv, such that v in H} is dense in H, and there is a constant $a > 0$, such that for all vectors v in H, $\|Sv\|$ is greater than or equal to a $\|v\|$ (Halmos, 1957). The last criterion is coined: T is bounded from below (Rubin,).

Verification showing that these two criteria hold when S is invertible follows since in this case, it is one-to-one and it is onto. As a consequence, note that the image of S is all of H, and it is therefore dense. Next, using $a = \|S^{-1}\|^{-1}$, then for all v in H, $\|v\| = \|S^{-1}Sv\|$ is less than or equal to $\|S^{-1}\| \|Sv\|$. So $\|Sv\|$ is greater than or equal to $\|v\| / \|S^{-1}\| = a \cdot \|v\|$.

Going the other way, to show that S is invertible given that the two properties hold, it must be shown that the following all hold. The first thing to show is that S is onto. Next, it must be shown that S is one-to-one. This is followed by showing that S^{-1} is linear. Finally, it must be proved that S^{-1} is bounded. It will be shown first that the range of S equals the codomain H, and so S is onto. Since S is dense, all that needs to be shown for being onto is that it is closed. So let w_n be a Cauchy sequence in the range of S. Let v_n be in the domain of S, such that $Sv_n = w_n$. It follows that since $\|w_n - w_m\| = \|Sv_n - Sv_m\|$ is greater or equal to a $\|v_n - v_m\|$, that v_n is also a Cauchy sequence in H. Consequently, v_n converges to v in H. Since the norm is continuous, it follows that $w_n \to Sv$, and so the range is closed. To show that S is 1−1, let $Sv_1 = Sv_2$, since $\|Sv_1 - Sv_2\| = 0$ and this quantity is greater than or equal to a $\|v_1 - v_2\|$. As a result, this shows that $v_1 = v_2$; accordingly, S is 1−1, and so S has an inverse. That S^{-1} is linear can be seen by using $S^{-1}(b\ {}_1w_1 + w_2) = S^{-1}(b\ Sv_1 + Sv_2) = S^{-1}\ [S\ (b\ v_1 + v_2)] = S^{-1}\ S\ (b\ v_1 + v_2) = I\ (b\ v_1 + v_2) = b\ v_1 + v_2 = b\ S^{-1}w_1 + S^{-1}w_2$. Finally, it will be shown that S^{-1} is a bounded operator. Use, $\|S\ v\| = \|w\|$ is greater or equal to, a $\|v\| = a\ \|S^{-1}\ w\|$, and so $\|S^{-1}\ w\|$ is less than or equal to $1/a\ \|w\|$.

Example 8.11:

The necessary and sufficient condition provided earlier for an operator T, on the Hilbert space H, to be invertible will be illustrated. If S and T are invertible, then $S \cdot T$ is invertible where the product is really function composition. For v in H, $(S \cdot T)(v) = S(T(v))$. It will be shown that $S \cdot T$ is bounded below and the range of $S \cdot T$ is dense in H. First note that T is bounded below means that $\|Tv\|$ is greater than or equal to a $\|v\|$. So, let $T(v) = w$ in H then since S is bounded from below, then $\|S(T(v))\| = \|S(w)\|$ is greater than or equal to b $\|w\| = b \|T\ v\|$, which is greater than or equal to a b $\|v\|$. Therefore, $S \cdot T$ is bounded from below. Next, show that $S \cdot T$ has dense range. This follows from basic definitions, S has dense range means that every point h in H is also in the range of $S = \{S(v) \mid v \text{ in } H\}$, or it is a limit point for points in this range. So there exist different cases. For instance, h is in the range of S; then since T also has dense range, it follows that $S \cdot T$ has dense range. On the other hand, let h be a limit point of the range of S. This means that every neighborhood of h contains a point w_n from the range of S. However, each of these points w_n lies in H. And because T also has dense range, every point w_n lies in its range or is itself a limit point. This means that h is also in the range of $S \cdot T = \{S(T(v)) \mid v \text{ in } H\}$, or it is a limit point of this range. As a result, $S \cdot T$ is invertible. Moreover, $(S \cdot T)^{-1} = T^{-1} \cdot S^{-1}$ since $(S \cdot T) \cdot (S \cdot T)^{-1} = S \cdot T \cdot T^{-1} \cdot S^{-1} = S \cdot I \cdot S^{-1} = S \cdot S^{-1} = I$.#

Example 8.12:

The set of all invertible operators $T: H \rightarrow H$, where H is a Hilbert space, form a nonabelian multiplicative group. Here, the multiplicative binary operation is function composition, so $(S \cdot T)(v) = S(T(v))$. The proof that $S \cdot T$ is invertible is given in Example 8.11. The unary operation is the inversion operation, and for T it is T^{-1}. The zero-ary element is the identity, I. The following equational identities hold:

1) Associative Law: $R \cdot (S \cdot T) = (R \cdot S) \cdot T$.
2) One Law: $I \cdot S = S \cdot I = S$.
3) Inverse Law: $S^{-1} \cdot S = S \cdot S^{-1} = I$.

The associative law holds. Let v be a vector in H, and then $T(v) = w$ is also in H; finally $S(w) = u$ is again in H as well as $R(u) = z$ in H. $(S \cdot T)(v) = S(T(v)) = S(w) = u$, and so $(R \cdot (S \cdot T))(v) = R(u) = z$. Going the other way, $T(v) = w$, $(R \cdot S)\ (w) = R(S(w)) = R(u) = z = ((R \cdot S) \cdot T)(v)$.#

Example 8.13: (Halmos, 1957)

Suppose that $\|T - I\| < 1$, it will be shown that T is invertible by illustrating that it is bounded below and the range of T is dense in H. Letting $a = 1 - \|I - T\|$, this shows that $a > 0$. If v is in H, then $\|T\ v\| = \|v - (v - T\ v)\|$ that is greater than or equal to $\|v\| - \|v - T\ v\|$, which is greater than or equal to $(1 - \|I - T\|)\|v\| = a\|v\|$. So T is bounded below by a. Also, notice that $1 - a = \|I - T\|$.

Next, by using the second sufficient condition for invertibility, it must be shown that the image of T, call it M, is dense in H. Consider any vector w in H. It will be shown that if $\delta = \inf \{ \|w - v\|$, where v is in M$\}$, then $\delta = 0$. Use contradiction, and assume that $\delta > 0$.

In this case, there is a vector z in M such that (1−a) $\|z-v\|<\delta$. Both z and T (w−z), as well as z+T (w − z), are elements of M. As a consequence, δ is less than or equal to $\|w-[(z+T(w-z))]\|$, because δ is the infimum. However, $\|w-[(z+T(w-z))]\| = \|(w-z)-T(w-z)\|$ is less than or equal to $\|I-T\|\,\|w-z\| = (1-a)\|w-z\|<\delta$. Resulting in a contradiction.#

Example 8.14:

Suppose that $\|S\|<1$; it will be shown that S−I is invertible by using the result from Example 8.13. Indeed, in the mentioned example, replace T by I−S. Substituting in that example for T shows that $\|(I-S)-I\|<1$, that is, $\|S\|<1$; this implies that I−S is invertible.#

Example 8.15:

For the bounded operator T, if it is bounded below, then its range is closed. This result follows by setting $w_n = T(v_n)$, n = 1, 2,... Letting w_n converge to w creates a Cauchy sequence (CS) in the image. If it is shown that w is also in the image of T, this means the image is closed. Using the bounded below criteria along with CS criteria, $\|T(v_n)-T(v_m)\|$ is greater than or equal to $a\|v_n-v_m\|$, showing that v_n is also a CS. This sequence converges to v. Using the continuity of T shows that T(v) = w, and so w is in the image of T.#

8.6 Spectrum in a Banach algebra

For an operator T in B (H, H), the spectrum of T is compact, that is, spT is compact. First spT will be shown to be closed, and then it will be shown to be bounded. As previously mentioned, the concept of inverse is extremely important in a Banach algebra A. This follows since the spectrum of an element T in A depends on the inverse existing or not. In particular, this spectrum is denoted by spT and is defined to be the set of all complex numbers c, such that (c I− T) is not invertible; here I is the identity element in A. It will be shown that for every element of A, the spectrum is never the empty set, and it is in fact a compact set in C. Since this is the field of complex numbers C, this means spT is always closed and bounded. The proof will utilize the complement of the spectrum C−spT.

The compliment of spT in C is called the resolvent set. The resolvent set, RsT, is defined to be the set of all complex numbers c, such that (c I − T) is invertible. Accordingly, the quantity $(c\,I-T)^{-1}$ will often be written as follows: 1/(c I−T). By showing RsT is open proves that spT is closed. Choose a point z in RsT, so (T−z I) is invertible. For λ in C, note that $\|I-(T-z)^{-1}(T-\lambda)\| = \|(T-z)^{-1}\,[(T-z)-(T-\lambda)]\|$, which is less than or equal to $\|(T-z)^{-1}\|\,|\lambda-z|$. The quantity $\|(T-z)^{-1}\|$ is bounded; therefore, making $|\lambda-z|$ small ensures that $\|I-(T-z)^{-1}(T-\lambda)\|$ can be made less than one. In this case, it is seen that $(T-z)^{-1}(T-\lambda)$ is invertible, see Example 8.13. So, for $|\lambda-z|$ small, it implies that $(T-\lambda)$ is also invertible. This means that RsT is open, because for all points λ within an arbitrary small radius about z in RsT, $(T-\lambda)$ is invertible. Consequently the spectrum of T, spT, is closed. To see that it is bounded, it will be shown that for λ in spT $|\lambda|$ is less than or equal to $\|T\|$. And since T is bounded, it follows that spT is compact. Now using proof by

contradiction, if $|\lambda| > \|T\|$, then $\|T/\lambda\| < 1$, and this implies that $(I - T/\lambda) = (\lambda - T)/\lambda$ is invertible, which is a contradiction.

The spectrum and resolvent sets are illustrated by simple examples and are examined in more depth in subsequent chapters.

Example 8.16:

Consider T = I, the identity element in A. Then, the only time (c I − I) is not invertible is when c = 1, that is, the single point 1. This is actually the point (1, 0) in the complex plane. So spI = {1}, which is a closed set in C. The point is called point spectrum and will be described subsequently. Also, in this case, the resolvent set is RsT, and it is the whole complex plane with the single point 1 missing.#

Example 8.17:

Consider the Banach algebra A, with carrier set consisting of all complex-valued continuous functions f, on the real interval [0, 1]. Now, c I − f is not invertible when the quantity c − f (z) = 0, for all z in [0,1]. Accordingly, the spectrum is spf = f ([0, 1]); it is the range of f. This is called a continuous spectrum; it is comprised of a continuous interval. The corresponding resolvent set Rsf = the whole complex plane with f ([0, 1]) removed. For instance, if f (z) = $e^{iz\pi}$, where z is in [0, 1], then Rsf = the whole complex plane with the upper semicircle centered at the origin of radius one removed along with the end points: that is, (1, 0), (−1, 0) are also removed. See Fig. 8.2 for an illustration of this continuous spectrum.#

Example 8.18:

This is an additional example of a continuous spectrum. It will also illustrate the concept of approximate eigenvalue, described rigorously later. Additionally, it involves the position operation. Indeed, in CL^2 [0, 1], the Hilbert space of complex-valued absolutely square integrable functions, M (f) (x) = x f (x). There is no point spectrum; however, the continuous spectrum is the closed interval [0, 1].

This will be shown first for points p in (0, 1) by considering a rectangular pulse h centered at the point p of width 2d and of height the square root of 1/(2d). In this case, $h = 1/(2d)^{1/2}\chi_{[p-d,p+d]}$. So squaring the height results in a rectangle whose area is one. Now consider $\|(M-p)(h)\|^2$. Then, for small values of d so that the rectangle pulse lies

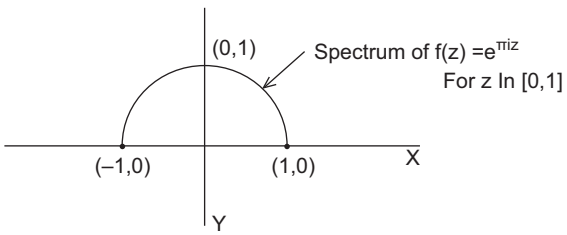


FIGURE 8.2 Continuous spectrum.

totally in (0, 1), this becomes the integral $\int_{p-d}^{p+d}((x-p)^2)/(2d)dx$. The result from the integral is $\|(M-p)\ (h)\|^2 = (d^2)/3$. As $d \to 0$, it will be seen in a subsequent chapter that p is an example of an approximate eigenvalue. This result holds true for every point p in the open interval (0, 1).

At zero and one, the same conclusion also holds by modifying h. That is, both zero and one are also in the spectrum of M. For instance, at zero, use $p = 0$, and a rectangle h of width d and height one over the square root of d, $d > 0$. In this case, $h = 1/(d)^{1/2}\chi_{[0,0+d]}$. Then $\|(M-0)(h)\|^2 = d^2$, as before letting $d \to 0$ shows that 0 is also an approximate eigenvalue. Accordingly, the close interval [0, 1] consists of the continuous spectrum for M.#

There exist three distinct types of spectrum. Two of them are illustrated earlier: the point spectrum and the continuous spectrum. The third is the residual spectrum. It is illustrated later even though it is of little importance in quantum.

To conclude this section, an additional example is provided illustrating the use of power series in Banach algebras.

Example 8.19:

Let v be in the Banach algebra A. The carrier set now consists of polynomials and formal power series expansions. Define w equal to the infinite sum representation of:

$e^v = I + v + v^2/2 + v^3/3! + v^4/4! + \ldots$. Since the series is absolutely convergent in this Banach algebra, w is also in A. Moreover, if u is also in A and the product commutes, that is, if $uv = vu$, then $e^{u+v} = e^u e^v$. (Muger, 2020).#

8.7 Ideals in a Banach algebra

For a Banach algebra A, the concept of an ideal in A involves a subset B. Whenever this subset is such that for v in A and w in B the product $v \cdot w$ is in B, then B is called a left ideal of A. The subset B is called a right ideal when $w \cdot v$ is in B, for all v in A. It is called a two-sided, or just an ideal when it is both a left ideal and also a right ideal. Fig. 8.3 illustrates a left and right ideal using a Venn-type diagram. When B differs from A and it differs from zero, the ideal is called proper. B is called maximal when it is proper, and B is not contained in another different proper ideal.

The importance of a closed and maximal ideal B, in a commutative Banach algebra A, is described next. In this case, B is a vector space, and the quotient space A/B is itself a Banach algebra. The resulting elements within A/B are equivalence classes of the form $[v] = v \cdot B = \{v \cdot w$, such that w in B$\}$ (Palmer, 1994). The norm in this Banach algebra is

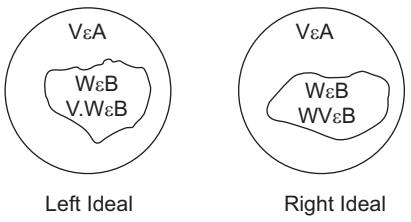


FIGURE 8.3 Left and right ideals.

given by $\|[v]\| = \inf\|v \cdot w\|$, for w in B. It follows from the CBS inequality that $\|[u]\ [v]\|$ is less than or equal to $\|[u]\|\ \|[v]\|$.

A special type of Banach algebra, namely a Banach* algebra, was described in Section 3.8. An illustration for the use of the star operation is given in the next example.

Example 8.20:

Let u and v be in A, where A is a Banach* algebra. Assume that $v = v^*$. For t in R, consider $w = e^{ivt}$ where $w = e^{ivt} = I + ivt + ((ivt)^2)/2 + \ldots$. This quantity is also located in A, since the series is absolutely convergent. Extend the properties of the * operation (1) and (2):

1) Involuntary: $(T^*)^* = T$
2) Additive: $(S+T)^* = S^* + T^*$

of a Banach* algebra in the limit, that is using the summation above; it follows that $(e^{ivt})^* = e^{-ivt}$. This implies that $\|(e^{ivt})\ e^{-ivt}\| = 1$, and therefore $\|w\| = 1$.#

A * homomorphism between two C* algebras, C1* and C2*, is a linear map L, such that:
L: C1* → C2* obeying the three equational identities:

1) Identity: L (I) = I.
2) Product: L (S T) = L (S) L (T), for S and T in C1*.
3) Involution: L (T*) = L (T)*.

Besides preserving the involution, the mapping is also contractive. That is, the mapping is continuous of norm less than or equal to one. So the norm of T in C1 is greater or equal to the norm of L (T) in C2. When L is one-to-one, the mapping is an isometry. Fig. 8.4 illustrates a homomorphism between two C* algebras.

8.8 Gelfand-Naimark-Segal construction

A representation for a C* algebra A consists of a star homomorphism and a Hilbert space H, that is, (L, H), where L: A → B (H), the set of all bounded operators on H. The representation (L, H) is said to be faith-full whenever L is 1−1; since L is linear, this is equivalent to the following: if L(f) = 0, then f = 0. A representation is called cyclic whenever there is a cyclic vector v, in H. This means the closure of (L(H) v) = H. It is equivalently said that the vector v is cyclic for B (H). Intuitively, a cyclic vector is such that the

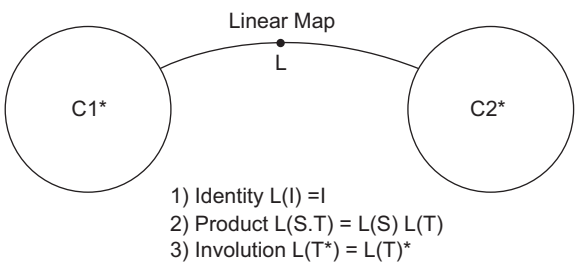


FIGURE 8.4 Graph for homomorphisms involving C* algebra.

repeated operator-type applications successively applied to this vector result in the whole space. An additional definition for a vector v is that it is separating. Here, v is separating whenever Tv = 0, for T in B (H) implies that v = 0. An important relationship between the concepts of cyclic and separating is due to the Gelfand-Naimark-Segal and is called the GNS construction. The relationship is in a C* algebra. In this case, a vector v is cyclic for B (H) when and only when v is separating for the commutant A′. The commutant was described earlier in Section 3.3. It is a subset of elements that commute with all other elements of A. Existence of cyclic vectors is one of the principal contributions in the GNS construction.

To say that v is separating for the commutant A′ means that for every element w of A′ commutes with v, that is, w v = v w. Here, A′ is the von Neumann algebra. In this section, the algebra should be thought as being a C* algebra just consisting of all bounded operators from H. Additional facts pertaining to the von Neumann algebra are described in the next chapter.

For v cyclic, the set of all such u v are dense in H. So, w u v = 0, and therefore, u = 0, on a dense subset of H. By continuity, it follows that u = 0 everywhere, and consequently, v is separating. Whenever A is also a von Neumann algebra, then a vector that is both cyclic and separating in A is also cyclic and separating in A′ (Sakai, 1971). An interesting example of a cyclic vector is given next.

Example 8.21:

As an example of a cyclic vector v, consider the carrier set $H = R^n$, for sort VECTOR. Use the usual inner product and matrix operations in this Hilbert space (He′lein, 2014). Let L be in B (H), and assume that L is self-adjoint. Then L can be diagonalized using a self-adjoint ON basis. Let the set of eigenvectors be {e1, e2,. . ., en}, along with the set of all real eigenvalues {c1, c2,. . ., cn}. Each eigenvalue ck is in the point spectrum of L, that is, it is in sppL. So it follows that L ek = ck ek, for all k above. In short, ck is an eigenvalue and ek is an eigenvector. Moreover, when all the eigenvalues are distinct, then v is cyclic; otherwise, it is not. To see this, represent L as a matrix; then by the Cayley-Hamilton theorem, L satisfies its characteristic equation: $L^n = \text{tr}(L)\ L^{n-1} + \ldots + (-1)^n \det(L)$.

Since H is finite dimensional, v is cyclic when and only when {v, Lv,. . ., L^{n-1} v} is a basis set for H. Using the eigenvectors mentioned earlier, v = d1e1+d2e2+. . .+dnen. Now, let powers of L operate on the vector v, L^k v = d1L^k e1+d2L^k e2+. . .+dnL^k en. Substituting into this equation with the eigenvalues results in L^k v = d1$c1^k$ e1+d2$c2^k$ e2 + . . . + dncn^k en, for k = 0, 1, 2,. . ., n − 1. These equations can be written as n by n matrix consisting of eigenvalues from L, multiplying an n by one column vector. This matrix multiplying the n by 1 column vector has tuples, dj ej, and is given below. It is followed by resulting n by 1 column vector with tuples, L^j v:

$$\begin{array}{cccccc} |1 & 1 & \ldots & 1| & |d1\ e1| & \\ |c1 & c2 & \ldots & cn| & |d2\ e2| & = \\ & \ldots & & & \ldots & \\ |c1^{n-1} & c2^{n-1} & \ldots & cn^{n-1}| & |dn\ en| & \end{array} \quad \begin{array}{c} \\ |L^0 v| \\ |L^1 v| \\ \ldots \\ |L^{n-1} v|. \end{array}$$

The n by n matrix above is a Vandermonde matrix, and it equals zero when and only when two columns are equal. So v is cyclic when and only when the eigenvalues are distinct. (Horn et al., 1957).#

The norm in a C* algebra is unique and is determined solely from its algebraic structure. Another GNS result is that every C* algebra is isometrically and involutionary isomorphic to a closed subinvolutionary algebra of bounded linear operators B (H), on a Hilbert space H. This is illustrated in Fig. 8.5. Moreover, when the C* algebra is separable, B (H) also can be separable. Additionally, for a commutative C* algebra, it is isomorphic to the algebra of complex-valued continuous functions vanishing at infinity. These functions are defined on a compact Hausdorff space. The GNS results specify representations of a C* algebra A, on a Hilbert space H, of bounded functions B (H). This is a star homomorphism h, h: A→B (H). Also h (I) = I. Moreover, for a state w in C*, there is a representation of A on a Hilbert space H with a unit vector v in H such that for all x in A it follows that w(x) = < v, h(x) v>. In general, the GNS construction using a pure state provides an irreducible representation; it cannot split into a direct sum of simpler representations. A mixed state usually leads to direct sums of independent representations.

When a C* algebra A is of finite dimension, it is isomorphic to a C* algebra involving a direct sum of nk by nk complex-valued matrices Mnk, k = 1, 2,. . ., N. The GNS construction shows that in quantum areas the C* algebra is generated by the observable. Additionally, when L is a representation of A on a Hilbert space H and v is a normalized cyclic vector, then f→<L(f)v, v> is a state in A. Recall that a state is a positive functional f on A, f: A→C; for all T in A, f (T*T) is greater than or equal to zero; and f(I) = 1. The operator T in A is also called a measurement operator; it is always self-adjoint. Additionally, the functional f often is an expectation operator. A subset J of A is a left ideal for A using J = {B in A | f (B*B) = 0}. The quotient space V = A/J can be made into an inner product space by using for B, D in A: <[B], [D]> = f (B*D). With completion, V becomes a Hilbert space (Sakai, 1971).

The GNS construction illustrates a strong connection between C* algebras and bounded functions on a Hilbert space. An example will show this relationship.

Example 8.22:

Consider the unital associative algebra A of all two by two matrices over the complex numbers. It is also a Banach algebra, and so with the adjoint operation for matrices in A, A becomes a C* algebra. The GNS construction provided below involves a pure state and

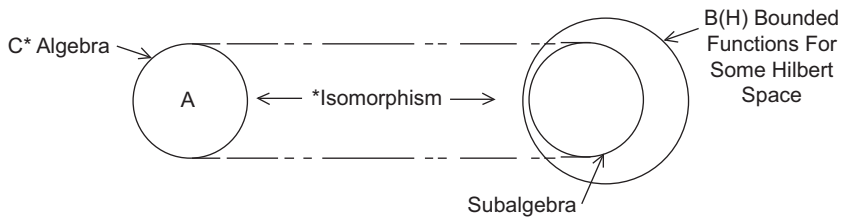


FIGURE 8.5 GNS construction between A C* algebra and a Hilbert space. *GNS*, Gelfand-Naimark-Segal.

will yield an isomorphism between A and a subalgebra of bounded functions for the Hilbert space C^2. To see this, use any element, that is, any complex-valued matrix M of A. Then a state f will be defined, f: A→C, where f (M) is the one, one entry within the matrix, for the matrix below, f (M) = a,

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}.$$

This is a state since it is linear, nonnegative, and f, evaluated at the identity matrix, is f (I) = 1. Nonnegativity follows since f (M*M) = a*a+c*c = $|a|^2 + |c|^2$. For a GNS construction, a sesquilinear form on A can be defined for matrices M and N in A, where N =

$$\begin{vmatrix} m & n \\ p & q \end{vmatrix}.$$

So let the pseudo inner product of these two matrices <M, N> = f (M*N) = ma*+pc*. This yields a sesquilinear form on A. Next, a subspace J of A will be found such that it is a left ideal of A. Use for J, J = {M in A, such that f (M* M) = 0}. Here, from the above, it follows that a = c = 0, and $\|M\|^2 = <M, M> = 0$. Accordingly, J is the subspace of A with the first column of M being zero. A quick multiplication of a matrix N from A times a matrix from J winds up with another matrix in J, thereby showing that J is a left ideal of A. In other words, for any N in A, the product of N and any matrix j from J results in another matrix j′ in J.

A Hilbert space arises for the GNS construction by setting H = A/J, the quotient space. An in-depth MSA development of the quotient space is provided in Section 10.1. Only basic facts from the quotient space are needed below. In any case, the associated inner product for equivalence classes within H is induced from the sesquilinear form above. So the inner product of two equivalence classes [M] and [N] is given by <[M], [N]> in H, which is equal to <M, N> = f (M*N). Note that for [M] in H, the coset [M] equals the sum of a matrix from A, plus a matrix with the first column equal to zero from J. These matrices are given below in order:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{vmatrix} 0 & n \\ 0 & q \end{vmatrix}.$$

As a representative for the equivalence class [M], we will use the following matrix as the coset leader:

$$\begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix}.$$

Notice that f (M*N) = ma*+pc*. This follows since M* and N are given in order:

$$\begin{vmatrix} a* & c* \\ 0 & 0 \end{vmatrix} \quad \begin{vmatrix} m & n \\ p & q \end{vmatrix}.$$

Next let L be a representation of A on the Hilbert space H, that is, let L: A→End(H) where L(N): [M]→[N M], so using N and [M] above L(N) (M) =

$$\begin{vmatrix} ma + nc & 0 \\ pa + qc & 0 \end{vmatrix}.$$

This follows since N, [M], and N • [M] are given below in order,

$$\begin{matrix} |m & n| & |a & 0| & |ma + nc & 0| \\ |p & q| & |c & 0| & |pa + qc & 0|. \end{matrix}$$

Moreover, note that the identity vector I in A corresponds in H to v = [I] =

$$\begin{matrix} |1 & 0| \\ |0 & 0| \end{matrix}$$

The vector v is a normalized cyclic vector in H. The vector v generates H. For any N in A, f (N) = < v, L (N) v>. This follows since f (N) = m, and L (N) v =

$$\begin{matrix} |m & 0| \\ |p & 0| \end{matrix}$$

and so f (v* L (N) v) = m. Accordingly, the Hilbert space H is isomorphic to C^2 with vectors in this space being acted upon by matrices such as N from the representation L (N). The GNS construction produced a Hilbert space, C^2 starting from a C* algebra, A of all complex-valued two by two matrices.#

8.9 Generating a C* algebra

For any subset S of a C* algebra A, which contains the identity, the smallest C* algebra containing S is called the C* algebra generated by S. It is denoted by B and is found by taking the intersection of all C* algebras that contain S. Moreover, for any element T in B, the spectrum of T in B equals the spectrum of T in A.

Example 8.23:

The criteria for generating a C* algebra B, using elements of a set of vectors S, from A, sound easy to do. However, noticing that the spectrum of T in B must also be in the spectrum of T in C* indicates there might be a difficulty in finding B. That indeed is the case. The C* algebra B has to be built up from vectors in the set S. All the closure operations for a C* algebra must hold for elements T in B. This means that all the vector space closure operations have to hold. So, linear combinations of vectors from S are in B. This means scalar multiplying vectors in S by all complex numbers, then adding them together in all possible ways must be in B.

Since a C* structure is an algebra, multiplication must also be closed. So, linear combinations of vectors from S as well as multiples of all these linear combinations must belong in B. Moreover, it is a Banach space, the norm of all these multiples must exist, and it has to be norm closed. Finally, the adjoint of these elements must also belong in B. In summary, elements of B are of the form, $\sum_{j=1}^{n} \left(c_j T_{j,1} \bullet T_{j,2} \bullet T_{j,3} \cdots T_{j,k}\right)$, for all nonnegative n and c_j in C, as well as limits as $n \to \infty$. Here, $T_{j,i}$ are in S or in S*. The summation mentioned earlier is often abbreviated using $\{T_{j,i}, T_{j,i}^*\}^n$, n = 0, 1, 2,. . . and $T_{j,i}$ are in S. Moreover, for n = 0, this means the identity operator is also in the algebra generated by S.

One of the most trivial examples is for two by two complex-valued matrices over the complex field. If S = {I}, then B = $\{I\}^n$; it consists of only {c I}, for all values of c complex valued. In this case, the result is all the diagonal matrices with the same scalar at each entry, that is, all scalar matrices.#

8.10 The Gelfand formula

Spectral radius formulas are given for bounded operators T, in Banach algebra: Here, r (T) is defined for T in B. It is r (T) = sup {| c |, such that c is in the spectrum of T}. It can also be found using Gelfand formula: Take inf $\|T^n\|^{1/n}$ for n = 1, 2, 3,. . ., or by taking the limit as n goes to infinity of $\|T^n\|^{1/n} = m$, nonzero. In any case, the series $\sum_{n=0}^{\infty} [T^n z^n]$ converges absolutely for | z | <1/m (Murphy, 1990).

Example 8.24:

As an illustration of the Gelfand formula, consider the carrier set consisting of all 2 by 2 matrices with the usual operations within the matrix Banach algebra. The matrix M is given below along with M^5 and M^7. The eigenvalues for M are −1 and 5; this constitutes the point spectrum of M.

$$M = \begin{vmatrix} 1 & 2 \\ 4 & 3 \end{vmatrix} \quad M^5 = \begin{vmatrix} 1041 & 1042 \\ 2084 & 2083 \end{vmatrix} \quad M^7 = \begin{vmatrix} 26041 & 26042 \\ 52084 & 52083 \end{vmatrix}$$

For the three matrices above, the corresponding roots of their norms are given along with two other instances. The spectral radius of M is 5, and for larger values of n in $\|M^n\|^{1/n}$, the value 5 is being approached: $\|M\| = 5.391$, $\|M^5\|^{1/5} = 5.053$, $\|M^7\|^{1/7} = 5.037$, $\|M^{15}\|^{1/15} = 5.017$, and $\|M^{25}\|^{1/25} = 5.010$.#

For any C* algebra, if T is self-adjoint then the spectral radius $r(T) = \lim_{n\to\infty} \|T^{2^n}\|^{2^{-n}}$. This follows using the self-adjoint property, $\|T^2\| = \|T^* T\| = \|T\|^2$. Then by mathematical induction, for any positive integer n, it follows that $\|T^{2^n}\| = \|T\|^{2^n}$, and so $\|T^{2^n}\|^{2^{-n}} = (\|T\|^{2^n})^{2^{-n}} = \|T\|$. This result can be used in showing that a norm in a C* algebra must be unique (Arveson, 1976).

The concluding structure is a Banach algebra. As mentioned previously, Banach algebras are understood to be unital and associative, in this document, unless specified otherwise. This is a famous example where the Banach algebra has no unit element.

Example 8.25:

This is an illustration of a Banach algebra that is commutative, but has no unital element. Consider all the complex valued, absolutely integrable functions in L^1. In this case, let BINE (f, g) = convolution of f and g. So, conv(f, g)(t) = $\int_0^{\infty}$ f(s) g (t−s) ds. There is no function in L^1 that is an identity. However, for generalized functions in the operational calculus, the delta function acts as an identity. Use of the delta function is illustrated in Section 6.3 for the tunneling effect.#

References

Arveson, W., 1976. An Invitation to C* Algebra. Springer-Verlag, 0-387-90176-0.
Besicovitch, A., 1954. Almost Periodic Functions. Dover Publications.
Bonsall, F., Duncan, J., 1973. Complete Normed Algebras. Springer, ISBN 978-3-642-65671-2.
Conway, J., 1990. second ed. A Course in Functional Analysis, vol. 96. Springer-Verlag, 978-0-387-97245-9.
Gelbaum, Olmsted, 2003. Counterexamples in Mathematical Analysis. Courier Co.
Halmos, P., 1957. Introduction to Hilbert Space and the Theory of Spectral Multiplicity. Chelsea Pub. Co., pp. 57–12834.
He'lein, F., 2014. Spectral Theory, 7. UMR CNRS, U Paris Didero, Paris, p. 1014.
Horn, R., Johnson, C., 1957. Topics in Matrix Analysis. Cambridge University Press, 1991. Chelsea Pub. Co, pp. 57–12834.
Lacey, H., 1973. The Hamel dimension of any infinite dimensional separable Bench space is c. Am. Math. Mon. 80, 298.
McCoy, N., 1929. On commutation formulas in the algebra of quantum mechanics. Transactons Am. Math. Soc. 31 (4).
Muger, M., 2020. Notes on the Theorem of Baker-Campbell-Hausdorff-Dynkin. BCHD.
Murphy, G., 1990. C* Algebras and Operator Theory. Academic Pres.
Palmer, T., 1994. Banach Algebras and the General Theory of *-Algebras. Cambridge U. Press.
Sakai, S., 1971. C* Algebras and W* Algebras. Springer, 3-540-63633-1.

CHAPTER

9

Von Neumann algebra

9.1 Operator topologies

Weak operator topology (WOT): For all v, w in H, $Tk \to T$, $k = 1, 2, \ldots$ WOT means that the convergence is defined with respect to the inner product, that is, as k goes to infinity, $<v, Tk\ w> \to <v, T\ w>$. On the other hand, strong operator topology (SOT), SOT: $Tk \to T$, means for all v in H, that is, k goes to infinity, the norm $\|Tk\ v - T\ v\| \to 0$. This type of convergence is likened to point-wise convergence in function spaces. The strongest convergence is in the norm topology: $Tk \to T$, which means that for k going to infinity the operator norm $\|Tk - T\| \to 0$. This type of convergence is likened to uniform convergence in function spaces. Norm convergence implies strong operator convergence, which itself implies weak operator convergence. For T and T_n in B (H), for $n = 0, 1, 2, \ldots$, the sequence of inequalities prevail; $|<v, (T_n - T)\ w>|$ is less than or equal to $\|(T_n - T)\ w\|\ \|v\|$, which is less than or equal to $\|T_n - T\|\ \|w\|\ \|v\|$. This shows that the norm convergence implies SOT, which in turn implies WOT convergence. The converse of these statements is false, and they are illustrated in the next few examples.

Example 9.1:

Consider the Hilbert space, $H = l^2$. Let the isometry T_n map a sequence $(v_0, v_1, v_2, \ldots)$ into a new sequence with zeros in the first n tuples, and right shifts the original tuples starting at the n + 1st tuple. That is, $T_n(v_0, v_1, v_2, \ldots) = (0, 0, \ldots 0, v_0, v_1, v_2, \ldots)$. Also consider the projection operator P_n, which just replaces the first n tuples of any vector w in l^2, with zeros. That is, $P_n(w_0, w_1, w_2, \ldots) = (0, 0, \ldots 0, w_n, w_{n+1}, \ldots)$. Then using any vectors w and v in l^2, the absolute inner product $|<w, T_n v>| = |<P_n\ w, T_n v>|$. But this quantity is less than or equal to $\|P_n\ w\|\ \|T_n v\| = \|P_n\ w\|\ \|v\| \to 0$, as $n \to \infty$, because $\|P_n w\|^2 = \sum_{k=n}^{\infty} |w_k|^2 \to 0$ as $k \to \infty$. So T_n converges to zero in the WOT. However, as mentioned earlier the operator T is an isometry, $\|T_n v\| = \|v\|$, for all n and any v in H, so it does not converge in the SOT. In general, weak convergence does not imply strong convergence.#

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00024-2

Example 9.2:

Strong convergence does not imply norm convergence. To see this, let the carrier set again be the Hilbert space, $H = l^2$. Also use the projection operator P_n, which just replaces the first n tuples of any vector w in l^2, with zeros. That is, P_n (w_0, w_1, w_2, ...) = (0, 0, ... 0, w_n, w_{n+1}, ...). $\|P_n\, w\|^2 \to 0$, as $n \to \infty$, since $\sum_{k=n}^{\infty} w_k^2 \to 0$. This shows SOT. In this case, there is no norm convergence. For any $m > n$, consider ($P_n - P_m$) (w_0, w_1, w_2, ...) = (0, 0, ... 0, w_n, w_{n+1}, ... w_m, 0, 0, ...). Then, there is a vector u in l^2, such that ($P_n - P_m$) u = u. As a consequence, $\|P_n - P_m\| = 1$, and therefore P_n is not a CS and so it does not converge in the norm.#

9.2 Two basic von Neumann algebras

A von Neumann algebra is often denoted as a W* algebra. It is a star algebra of bounded operators on a Hilbert space, which is closed under the WOT and contains the identity operator. Two famous examples of a W* algebra are the ring of essentially bounded measurable functions, $L^{\infty}(R)$. It is a commutative von Neumann algebra consisting of elements acting as point-wise multiplication on the Hilbert space of L^2 functions. The second example consists of all bounded operators on a Hilbert space of dimension greater than one. This algebra is noncommutative. Proofs of these can be found in Jones (2010), also, see Nelson (2017).

Brief remarks will be given only to illustrate some concepts needed in the proof of the first result mentioned earlier. This proof shows that the structure is a von Neumann space. First, a sigma finite measure (X, Ω, μ) is assumed in the proof. This structure is defined in Appendix A.2. Also needed is the set $M = L^{\infty}(R)$, which is the normed vector space of essentially bounded measurable functions on R, with essential supremum norm $\|f\|_{\infty}$. This norm is equal to inf {N, nonnegative N, such that | f(x) | is less than or equal to N, everywhere except on a set A in X, such that A has measure zero, that is, μ(A) is zero}. An example of this norm follows.

Example 9.3:

As an instance of $\|f\|_{\infty}$ norm, consider the function f, where f: R $\to$ R, and f(t) = zero except that f(t) = 1/ t in the open interval (2, 4), along with the assumption that f(4) = 5. Then $\|f\|_{\infty} = 1/2$, because point four is of measure zero and does not matter using this norm.#

Also indicated in the proof of the von Neumann algebra involving multiplication on the L^2 space is the use of bounded functions in L^2. Here, an embedding of $L^{\infty}(R) \to B(L^2(X, \mu))$ is established with an image of a von Neumann algebra in B (L^2 (X, μ)). In this case, μ is the Lebesgue measure. This space consists of all functions that are bounded and also absolute square integrable. Not all functions in L^2 are bounded, as the following example illustrates.

Example 9.4:

Use the sequence of functions, on the real line, $f_n(t) = n^{1/2}$, for t in the interval, (n, n + 1/n), n = 1, 2, 3,. . ., and zero elsewhere in R. The functions $f_n(t)$, are in B (L^2 (X, μ)). However, the limit of this sequence as n goes to infinity is not in B (L^2 (X, μ)), even though it is in L^2 (X, μ), with value one.#

9.3 Commutant in a von Neumann algebra

The von Neumann algebra A consists of a unital star subalgebra of the bounded operators in a Hilbert space. So, I is always in A along with the adjoint of any element of A. The von Neumann algebra is closed in the weak operator topology. This differs from a C* algebra. A C* algebra is closed in the norm topology. The commutant A′ is also a von Neumann algebra. The commutant A′ was described and illustrated in an earlier section involving matrices. Here, A′ = {v in B (H), such that v w = w v for all w in A}. Additionally, the intersection of A and A′, Z(A), is a von Neumann algebra called the center of A. Whenever Z(A) = A, A is said to be abelian. The double commutant is A′′ = (A′)′. For T, a bounded operator in Hilbert space H, a double commutant can be found from the generating set, $\{T, T^*\}^n$, n = 0, 1, 2,. . . . This is a subset S of the C* algebra A, which contains the identity $T^0 = I$. It is the smallest C* algebra containing S and is found by taking the scalar products, sums and products as well as sums of products involving T and T*. See Section 8.9 for an example of generating sets. The von Neumann double commutant theorem states that when A is a subset of B (H), which is unital and a star subalgebra of B (H), then A is strongly dense in A″ (Dixmier, 1981).

Example 9.5:

Consider the double commutant of T in the Hilbert space $H = C^2$ given below. It results in all two by two complex-valued matrices over the complex field C. Here, T is given along with T*.

$$\begin{matrix} |0 & 1| \\ |0 & 0| \end{matrix} \quad \begin{matrix} |0 & 0| \\ |1 & 0| \end{matrix}.$$

Together with the identity I, the quantity $\{T, T^*\}^n$ represents all scalar multiples, sums, products, and linear combinations of these quantities, as well as the strong operator closure of all these entries. This results in the algebra of two by two matrices over the complex numbers.#

The trace Tr, in the von Neumann algebra V, is a subalgebra of B(H) for positive self-adjoint operators. It involves V + = {A, such that A in V and <v, Av> is greater than or equal to zero for all v in H.}. The trace on V (Nelson, 2017) is Tr: M + → [0, ∞], such that:

1) Additive: Tr(A + B) = Tr(A) + Tr (B), for A and B in V + .

2) Homogeneous:: Tr (c A) = c Tr (A) for A in V + , c greater than or equal to zero.
3) Faithful: Tr (A) = 0 implies A = 0.
4) Normal: For a monotonic increasing sequence Ak in V + converging in operator norm to A also in V + , then Tr (Ak) converges to Tr (A).
5) Semifinite: Tr (A) = sup {Tr (B), such that B is in V + , Tr (B) is finite, A − B is greater than or equal to zero} for all A in V + .
More generally, for A not necessarily positive
6) Tracial: Tr(A A*) = Tr (A* A) for A in V.

Example 9.6:

Let the von Neumann algebra be V = B(H) and A in V + , then a trace function for V utilizes an ON basis for H, {en}, n = 1, 2,. . . . Tr (A) = sum $\sum_{n=1}^{\infty} <$ en, Aen $>$. Any other trace is of the form c Tr (A) for c $>$ 0.#

Example 9.7:

Consider the Hilbert space H = C^n, and let Mn(C) be the set of all nxn complex matrices constituting the von Neumann algebra B(H). The trace Tr: Mn(C) → C, where for A in Mn (C), then Tr(A) = sum $\sum_{j=1}^{n}$ Ajj, where the Ajj are diagonal elements in A. Using the standard basis {e1, e2n, . . ., en} in Mn(C), the Tr(A) = sum, $\sum_{j=1}^{n} <$ ej, Aej $>$. For any ON basis, {f1, f2, . . ., fn} in Mn(C), the Tr(A) = $<$ fj, Afj $>$. Let U be a unitary matrix sending column vectors ej to fj, that is, Uej = fj, j = 1, 2,. . ., n. Then Tr(A) = sum, $\sum_{j=1}^{n} <$ fj, Afj $>$ = sum $\sum_{j=1}^{n} <$ Uej, AUej $>$ = Tr(A) = sum $\sum_{j=1}^{n} <$ ej, U * AUej $>$ = Tr(U * AU) = Tr(AUU*) = Tr(A).#

9.4 The Gelfand transform

For any Banach algebra A, over C, let Δ be the space of all multiplicative linear functionals on A. These are also called the characters of A. The meaning is that the linear functional T is a nontrivial algebra homomorphism, T: A → C. It has the property that T(v · w) = T (v) T (w), for all v and w in A. The set of all these multiplicative linear functionals on A form a locally compact Hausdorff space in the weak * topology. Additionally, this set is compact, because it is assumed that the identity element I is in A. The compact Hausdorff space Δ(A) is called character space. Let C(Δ(A)) be the algebra of all complex-valued continuous functions in Δ. The Gelfand transform is G: A → C(Δ), where v → G (v), G (v) is in C(Δ) and is defined as G(v) = v̂, where x̂(T) = T(v), for all T in Δ.

The algebra of complex-valued continuous functions in Δ that vanish at infinity is a subalgebra of C (Δ) and is denoted by C_0 (Δ). The range of the Gelfand transform is contained in C_0 (Δ). The GNS construction is introduced in Section 8.8. It is extended to all commutative C* algebras A, over C with the help of the Gelfand formula. In this case, A is * isomorphism to C_0 (X) for some compact Hausdorff space X. Additionally, the Gelfand

transform G is a * isomorphism between A and C_0 (Δ). When there is no identity in the C* algebra, the Hausdorff space need to be only locally compact. This is a result of the Banach-Alaoglu theorem (Narici and Beckenstein, 2011). The Gelfand transform is the backbone of numerous transform methods involving absolutely integrable functions. A couple of examples follow.

Example 9.8:

Consider, the set A consisting of all absolutely integrable functions on the real line. Integration is with respect to the Lebesgue measure. This space is denoted by $L^1(R)$. Using point-wise addition and multiplication involving the convolutional operation, A becomes a C* algebra without an identity function. In any case, it is also called a group algebra. The convolution operation is quite similar to the ones defined using bound matrices; however, an integral replaces the summation signs in the present situation. For f and g in $L^1(R)$, the multiplication is $f \circledast g\ (t) = \int_{x=-\infty}^{\infty} f\ (x)\ g\ (t - x)\ dx$. The Gelfand transform in this case is the Fourier transform. For f in $L^1(R)$, this transform results in a function F (ω) with complex values, namely, $F\ (\omega) = \int_{x=-\infty}^{\infty} f\ (x)\ e^{-ix\omega}\ dx$ It is uniformly continuous and has the Riemann Legesgue property, which is $F\ (\omega) \to 0$ as $|\omega| \to \infty$. This is consistent with the Gelfand transform resulting in a function in C_0 (X). That is a continuous function that goes to zero as the argument tends to plus or minus infinity.

A more in-depth treatment of the Fourier transform and its properties are provided in Example 19.10, in the context of reproducing kernel Hilbert spaces.#

Example 9.9:

Let A consist of all absolutely integrable functions on the nonnegative real line. Again, integration is with respect to the Lebesgue measure. This space is denoted by $L^1(R+)$. Using point-wise addition and multiplication being the convolutional operation, A again becomes a C* algebra without an identity function. The convolution of two functions in this space is $f \circledast g\ (t) = \int_{x=0}^{\infty} f\ (x)\ g\ (t - x)\ dx$. The Gelfand transform in this situation is the Laplace transform involving the variable s. This scalar is complex-valued with real part greater than or equal to zero. Here, $L\ (s) = \int_{t=0}^{\infty} f\ (x)\ e^{-st}\ dx$. The result is an analytic function in the region of absolute convergence.#

References

Dixmier, J., 1981. Von Neumann Allgebras. North Holland Pub. Co.
Jones, V.F.R., 2010. Von Neumann Algebras. U. Ca. Berkeley.
Narici, L., Beckenstein, E., 2011. 978–1584888666 Topological Vector Spaces. CRC Press.
Nelson, B., 2017. Von Neumann Algebras. U. Ca. Berkeley.

This page intentionally left blank

CHAPTER

10

Fiber bundles

10.1 MSA for the algebraic quotient spaces

In the MSA description of an algebraic quotient space, there exist four sorts; these are VECT, SUBV, SCALAR, and COSET. As usual, SCALAR refers to the field, for instance, the real or complex numbers, as well as the quaternion skew field. VECT refers to a vector space, and SUBV indicates a fixed subspace within VECT. Finally, COSET is the name of all vectors within the quotient space. Actually, the vectors v in COSET will consist of equivalence classes and are indicated by [v]. The quantity v within the brackets can be considered as a coset leader and will be described more thoroughly later. The actual signature sets are exactly as before when describing vector spaces and scalars. Additionally, all the equational identities for a vector space also have to hold. In order to compress notation, represent the sorts:

VECTOR by V
SCALAR by S
SUBV by N

Utilize symbols representing operator names:

V-ADD by $+$
V-MINUS by $-$, In the following, $-$ is used as a binary operation instead of writing $+(-)$.

An equivalence relation as described in previous chapters is an RST relation. Here it is defined on V by saying v is related to w whenever $v-w$ is in the subspace N of V. The subspace N is arbitrary and not empty. The relation can also be described as follows: v is equivalent to w whenever $v=w+n$, where n is in N. It is denoted by the equivalence sign, $v\sim w$. This motivates the notation for a coset, $[v]=\{v+n$ where n is in $N\}$, and in this case, from above, $[v]=[w]$. Moreover, $[0]=\{n$ is in $N\}$. So, all vectors in N act like ZERO. The quotient space V/N is defined as the set of all equivalence classes caused by the relation $\sim$ on V. It is also the space consisting of all the cosets, that is, $V/\sim$. The quotient space is itself a vector space when addition and scalar multiplication is defined as given below (Halmos, 1974). However, first make an abuse of notation by letting:

For COSET, use [v], [w].
For SCALAR, use a in S.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00004-7

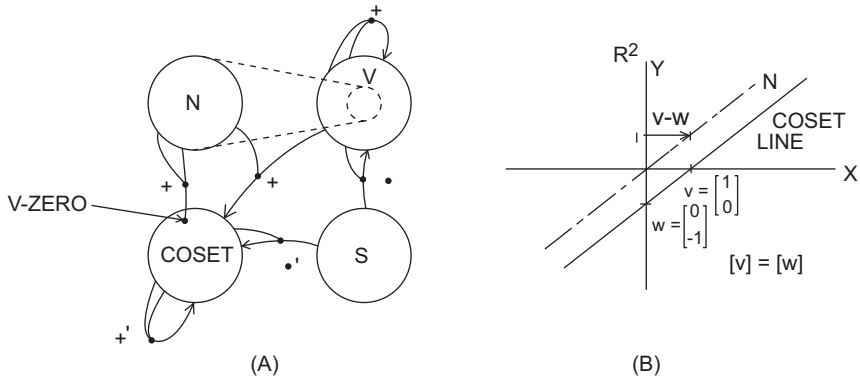


FIGURE 10.1 General and specific algebraic quotient spaces. (A) Mappings in quotient space, (B) quotient space, line in plane.

For V-ADD, use $+$, for elements or vectors in V.
For S-MULT in V, use $\cdot$; this multiplication involves scalars in S and elements in V.
For V-ADD, use $+'$, for elements in COSET.
For S-MULT, use $\cdot'$, for scalars in S and elements in COSET.

Two closure operations are provided next. Also, see the polyadic graph in Fig. 10.1A. Here, all four sorts appear; however, only some polyadic arrows are shown for V-ADD and S-MULT involving V, N, COSET, and S. The subspace N should be in the interior of V, but it is depicted separately in order to illustrate the operational names in a clear fashion. In any case, the closure requirements are as follows:

1. Homogeneous: $a \cdot' [v] = [a \cdot v]$.
2. Linear: $[v] +' [w] = [v + w]$.

In this figure, it is depicted that when two vectors in the subspace N are added together, the result is the zero equivalence class in the quotient space. So the two vectors map into [0] in COSET. When a vector from N and another vector from V get added together, the result is a coset in COSET the quotient space. Also illustrated in this diagram is a scalar a; multiplying an element [v] in the COSET, that is, $a \cdot' [v]$, the product is in COSET. Additionally, the result is given by criteria (1) above $a \cdot' [v] = [a \cdot v]$. Also in this diagram, it can be seen that two cosets, say [v] and [w], in the quotient space are added; the sum is again a coset. The sum can also be found by referring to criteria number (2) above $[v] +' [w] = [v + w]$.

Example 10.1:

Use the carrier set R^2 for the sort VECTOR, and use R for the sort SCALAR. Additionally, let the carrier set for subspace sort SUBV consist of vectors on a 45-degree line in the x, y plane. In Fig. 10.1B, there is an illustration of SUBV, the subspace N, along with a typical coset, consisting of a parallel line, to the right of N. Two vectors on this line are $v = (1\ 0)'$ and $w = (0\ -1)'$. The prime in this expression indicates matrix transpose, so

that these two vectors are column vectors. In any case, the difference between these two vectors is a vector in N. This vector is illustrated in the diagram, and the difference vector is located at (1 1)′. In this case, cosets are all affine spaces, that is, lines parallel to N.#

10.2 The topological quotient space

In topology, the quotient space is also defined by using equivalence classes as in the algebra described in the previous section. The set X used in the topological quotient space can be any set; it need not be a vector space. The topology Tx consists of a collection of subsets of X, called open sets. They are such that the empty set and the whole set X are in the topology along with the finite intersection as well as the arbitrary union of open sets. Repeating the requirements, Tx determines the sets that are open in the topology on X. It is such that:

1. Both X and the empty set ϕ are in Tx.
2. The union of any number of sets in Tx must be in Tx.
3. The finite intersection of any sets within Tx must also be in TX.

The quantity Tx is called the topology on X, and the topological space is denoted by (X, Tx) (Munkres, 1999). From an MSA perspective, Tx can be viewed similarly to MSA describing a sigma field, as indicated in Appendix A.2. Namely, in the present case, there exist two sorts, POWX, and OPEN. For a concrete example, POWX is the name for the power set 2^X. Additionally, OPEN is symbolic for all the open sets in the topology. Finally, a single signature set containing a unary operation Tx finalizes the high view of a topological space in the MSA methodology. Here, Tx: $2^X \rightarrow$OPEN, so Tx is a partial identity function in this case. It selects which subsets of X are to be open always obeying the constraining equations (1) − (3) above.

The quotient space (Y, Ty) is a topological space obtained from another topological space (X, Tx) using an equivalence relation along with the quotient topology Ty. The latter topology is the finest topology making the onto canonical projection map p: X→Y continuous. So, a subset U is open in Y when and only when p^{-1} (U) is open in X. This defines the topology Ty on Y. It does not say that p is an open map, that is, if V is in Tx, which is an open set in X, then p(V) need not be an open set in Y. The process of making p continuous is the opposite of that used in fiber bundles. In the latter case, it is the coarsest or initial topology. Here, it is the final topology. Again, the quotient map, p: X → Y, is always:

1. Onto.
2. Continuous.
3. For any set A in Y for which p^{-1}(A) is open in X, then A must be open in Y.

The criteria in number (3) always hold for a canonical mapping. Utilizing the quotient topology, whenever an open set A in X is a union of inverses of equivalence classes, then A is called saturated. Equivalently, A is saturated with respect to p, which means that p^{-1} (U [.]) = A. In this case, p(A) is open in Y, so for this situation p is an open mapping, but it is not an open mapping in general.

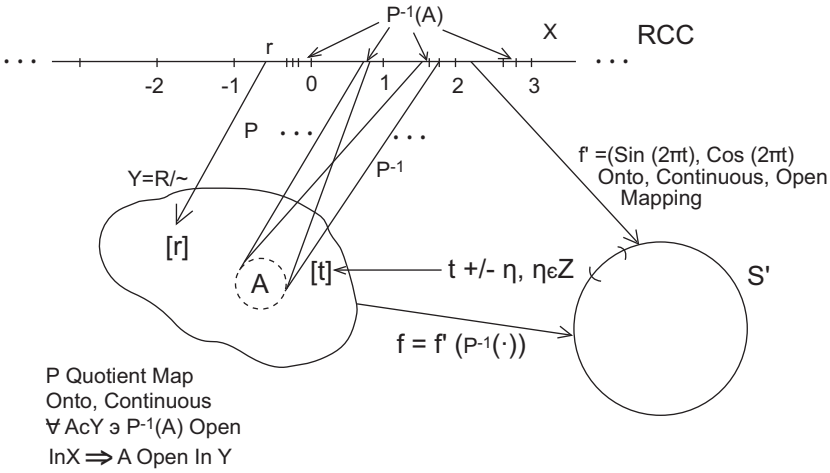


FIGURE 10.2 Homeomorphism of quotient space of reals.

Example 10.2:

Use X = R with the usual topology. Let the points r and s in R be related, r ~ s, iff the difference r-s is an integer. As a consequence, the quotient space Y = R/ ~ will be homeomorphic to the unit circle. In this case, the quotient space Y = R/ ~ = {{0+ / − n, for n in Z}, {t+ /− n, for n in Z}, such that t in (0, 1)}} = {[0], [t], such that t in (0, 1)}. Refer to Fig. 10.2. Here, the map p takes a real number r and maps it into equivalence classes [r]. In the quotient space Y, an open set A is illustrated in this diagram. Also in this diagram, the inverse operation p^{-1} (A) maps into open sets in R. These infinite number of sets are open intervals of the same size and disjoint, all an integer apart.

Let f′: R → S^1, where S^1 is the unit circle, and use the parameterization, f′ (t) = (sin(2πt), cos(2πt)). This function is continuous, and moreover, it is an open mapping. The latter condition is most easily seen from complex analysis. Nonconstant holomorphic maps are open maps, and when restricted to the real axis, it is an open mapping. Additionally, this mapping is a local homeomorphism, which is always an open mapping (Rudin, 1966). Define f: Y → S^1, by f([t]) = f′ (t). The function f′ at all integers always provides the same value, thereby showing that f is well defined. In the figure, note that f′ (.) = f (p(.)); this equation will be used later.

The function f will be shown to be a homeomorphism. It is bijective since it is onto and 1−1. Onto follows by choosing any point, in S^1; it is of the form (sin(2πt′), cos(2πt′)). The value of t′ used here is a point t′ in R. However, [t′] in Y corresponds to all points t′ + n in R, for n in Z showing f is onto. For 1−1, if $[t_1]$ does not equal $[t_2]$, then t_1 does not equal to t_2 + n, where n is an integer. As a consequence, sin(2πt_1) does not equal sin(2π (t_2 + n)) = sin(2πt_2); accordingly, f($[t_1]$) does not equal f($[t_2]$). Next, since p is a quotient mapping, it is continuous. Also, f′ is continuous, and so f'^{-1}(U) is open for U open. Since f'^{-1} = (f (p))$^{-1}$ = p^{-1}(f^{-1}), applying this to an open set U in S^1, f'^{-1}(U) = p^{-1}(f^{-1})(U)) is open, which implies that f^{-1}(U) is open since U is open in Y when and only when p^{-1} (U) is

open in X; this shows that f is continuous. For a homeomorphism, the only thing left to show is that f is an open mapping. However, this does follow since f′ is an open mapping. Since f (.) = f′ (p^{-1} (.)), apply this mapping to U an open set in Y. Then, f (U) = f′ (p^{-1} (U)), this is an open mapping since p is continuous, and f′ is an open mapping. Thus, the homeomorphism of Y and S^1 is established.#

Example 10.3:

Consider the unit interval X = [0, 1] with the subset topology Tx, induced from the real line. Use the equivalence relation on X, 0 ~ 1. This means that 0 and 1 are glued together. Let q: X → [0, 1]/ ~ = Y be the canonical projection map. Here, [0, 1]/ ~ = {{0, 1}, {t}}, and {t} are singleton sets for every t in (0, 1). Accordingly, for any t in (0, 1), q(t) = [t], but for 0 or 1, q(0) = q(1) = [0] = [1]. The quotient topology is employed; here, U is open in Y iff q^{-1}(U) is open in X. Since 0 and 1 are the same, it seems that this structure is like a circle. In fact, Y is also homeomorphic to S^1. To see this first, let f′: X → S^1, where f′ (t) = (sin(2πt), cos(2πt)); it is a continuous map. Define f, where f([t]) = f′ (t), for t in (0, 1) and f([0]) = f′ (0) = f′ (1). Refer to Fig. 10.3.

To show that f: Y → S^1 is onto, take a specific point in S^1, say c = (sin(2πt′), cos(2πt′)), then the point t′ in X is such that f′ (t′) = c. This means that [t′] in Y is such that f ([t′]) = c. To see that f is 1−1, if [t_1] does not equal [t_2], then t_1 differs from t_2, and sin(2πt_1) is not equal to sin(2πt_2), because the only time they can be equal is when t_1 and t_2 differ by an integer, that is at the end points of [0, 1]. Since they are the same in the equivalence class Y, 1−1 is proven. The homeomorphism for f follows by using a theorem from topology (Dugundji, 1975). Here, the continuous bijective function f from a compact space Y to a Hausdorff space is a homeomorphism. Note that S^1 is a Hausdorff space because R^2 is a Hausdorff space. The quotient space Y = R/ ~ is compact. This can be seen since it is the onto the continuous image of a compact set, namely X is compact, and the homeomorphism is shown.#

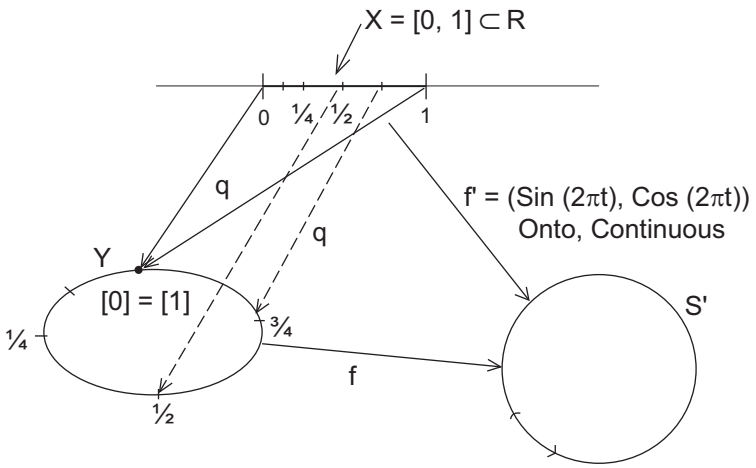


FIGURE 10.3 Homeomorphism involving circular interval.

Example 10.4:

The mapping q: (0, 1]→S^1, where q(t) = (sin(2πt), cos(2πt)) is not a homeomorphism, since q^{-1} is not continuous. To show this, consider the distance in R^2 between the point (0, 1) and (sin(2πt), cos(2πt)). It is the square root of $[\sin(2\pi t)^2 + (1 - \cos(2\pi t))^2] = 2(1 - \cos(2\pi t))$. So for $2(1 - \cos(2\pi t))$ less than delta, for t about zero, this should imply that |t−1| is less than epsilon, but there is a gap between t and one. More quickly, to show that they are not homeomorphic, use the fact that S^1 is compact, but (0, 1] is not compact in R. #

Example 10.5:

Again consider the real line with the usual topology. This time, consider Y = R/∼, where for all x and y in Z, x∼y means x = y. However, for all nonintegers, that is, for all x not in Z, x∼x. These points are singletons. The resulting quotient space is illustrated in Fig. 10.4, along with the quotient map q. The resulting structure is an excellent example of a nonfirst countable space. The point [0], which represents all the integers, does not have a countable basis. Let {Un}, n in N, be a collection of neighborhoods of [0]. It will be noticed that the open set U defined below will not contain any Un. This result will be seen by finding open intervals U′n containing individual integers n = 0, + / − 1, 2, ... where the Un′ are proper nonempty subsets of q^{-1}(Un). Letting U = q (union of all Un′), then no Un can be in U, for if it were, then Un′ cannot be a proper subset of q^{-1}(Un).#

10.3 Basic topological and manifold concepts

Manifolds M are always special topological spaces of the Hausdorff type. These spaces are such that every distinct two points must each be contained in distinct open sets disjoint from one another. Moreover, the Hausdorff space must everywhere be locally

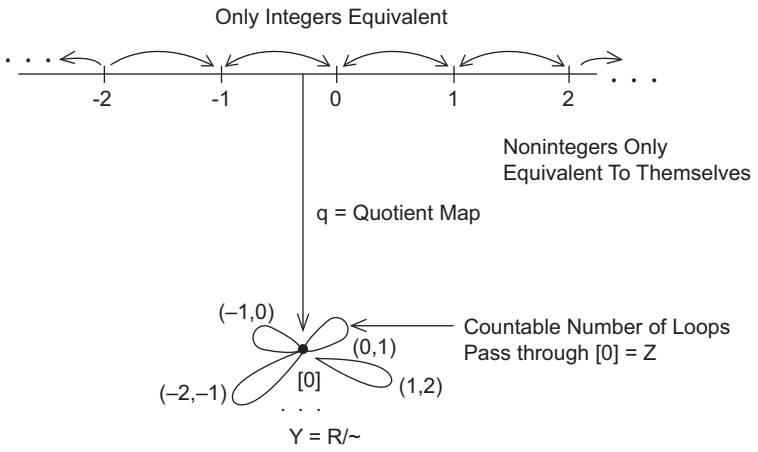


FIGURE 10.4 Nonfirst countable space.

homeomorphic to R^n. So, this means it appears to be locally like a Euclidean space, even though the overall structure might be highly nonlinear. A homeomorphic map is a bicontinuous, one-to-one onto structure-preserving map. Finally, most of the time, M is second countable, which means there exists a countable basis that is a countable union of open sets. This feature is assumed from here on. Second, countable implies first countable; the latter topological space needs to only have local countable bases. An illustration of a nonfirst countable space is given in Fig. 10.4. This example is pathological; moreover, Euclidean space is both a Hausdorff space and a second countable space.

A manifold M is always associated with an atlas A. This is a collection of open sets called charts, denoted by Ua. These charts describe regions within the manifold. Additionally, their union covers M. So in an atlas every point of the manifold must have coordinates within a chart. Accompanying these charts within an atlas are always functions, fa. The functions are such that fa: $M \to R^n$, where fa and fa^{-1} are homeomorphisms. The chart is called the coordinate domain, and fa is called the coordinate map. Often an atlas along with the corresponding manifold is called smooth when $fbfa^{-1}$ has k continuous derivatives, $k = 1, 2, \ldots$. Accordingly, they are in C^k. More often than not, smooth means infinitely differentiable. Within a chart, the various group structures will be viewed as an n-dimensional vector space with local coordinates. For the local coordinate system, the identity plays a special role to be seen in later chapters. Usually, a global coordinate system for the whole manifold is not feasible. For instance, using the surface of a sphere, only local representations can be exhibited.

Two charts Ua and Ub, with nonempty intersection, are in the manifold M. They are within a fixed atlas and are compared using a transition function T. See Fig. 10.5. For an atlas to be usable, the overlapping regions must not differ too much. The transition functions such that if fa: $Ua \to R^n$ and likewise fb: $Ub \to R^n$, then $Tab = fb\ fa^{-1}$, that is, Tab: fa (Ua intersect Ub) $\to$ fb (Ua intersect Ub). Note that T is also a homeomorphism since both fa and fb are homeomorphisms. There can be several atlases for a manifold. A maximal atlas is one which is not a proper subset of another atlas. For every smooth atlas A on M, there exists a unique maximal atlas on M. A differential structure, which is a globally defined differential structure, on a manifold is a maximal atlas (Lee, 2006).

A connected topological space T cannot be the disjoint union of two or more nonempty open sets. When a space is not connected, in these cases, there are subsets within the space

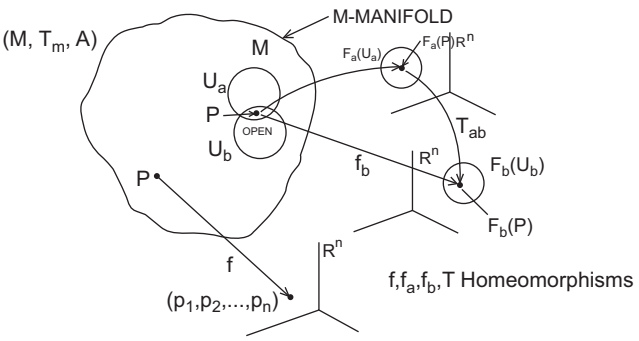


FIGURE 10.5 Manifold with two charts and transition mapping.

that are connected. These are called connected components. Connected components are maximally connected subsets of a topological space T. A different concept of connectivity is path connection. A path-connected space T means that there is a continuous function f: [0, 1| → T, such that for x, y in T, f(0) = x, f (1) = y. The path connected implies connected.

A topological space is compact whenever every open cover has a finite subcover. A maximal compact subgroup is a compact subgroup that is the largest. In R^n, compactness can be stated more easily: Here, every closed and bounded subset is compact; this is the Heine-Borel theorem. Also, compact manifolds are of special importance since there exist atlases with only finitely many charts. This is because in a compact group, every open cover has a finite subcover and charts are open sets.

10.4 Fiber bundles from manifolds

Fiber bundles are frequently comprised of manifolds held together by continuous functions. The manifolds always include the base B, and the fiber F, which is intuitively like strands or sticks of arbitrary length. Together, they create the total space E. For instance, if B is the circle S^1, place match sticks or line segments of intervals (0,1), perpendicular to and upright surrounding the circle. This represents the fiber F, and accordingly, a cylinder E is produced. It appears like a cylinder or a can with walls constituting the fiber and the bottom being the base. Here, the total space E = B x F. A different structure results, called the Möbius band by using a similar technique. Begin with the same base B, the circle. Again use sticks or intervals (0,1) surrounding the base circle to create the fiber F. However, now employ a gradual twist of 180 degrees while placing the line segments or sticks on the base. This is illustrated in Fig. 10.6. The twist is not performed at a single point of the circle; rather, it is uniformly distributed throughout the circle. However, the

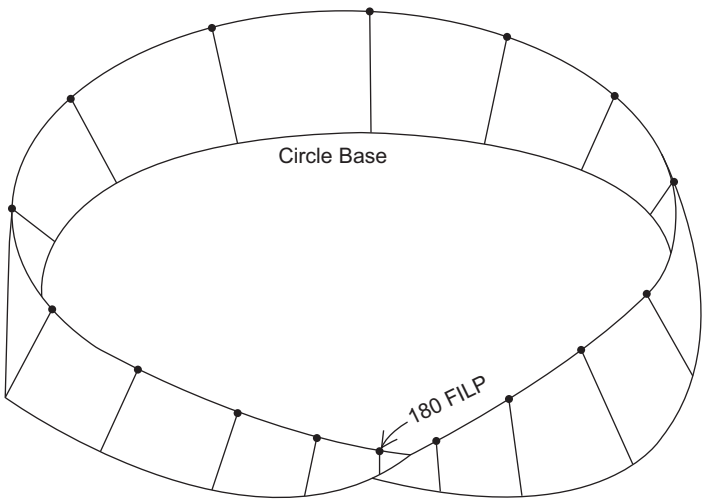


FIGURE 10.6 Möbius stick figure.

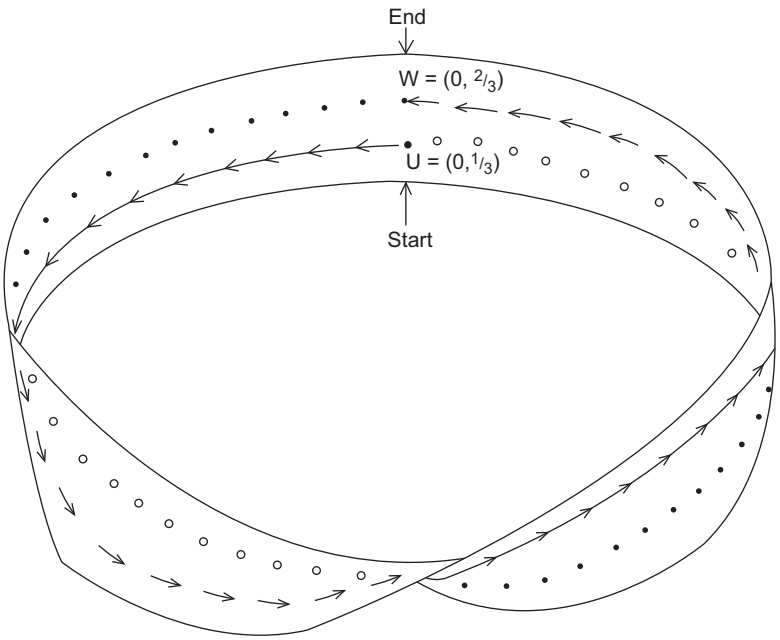


FIGURE 10.7 Möbius strip.

diagram provided does not illustrate this very well. The twisting can be viewed as a twisting of the local charts. The two structures, the cylinder, and Möbius strip are similar. They utilize the same base and about the same amount of fiber for construction. But the end results are quite different due to parameterization.

The Möbius structure is locally a product space, but it is not globally a product space as is the cylinder. Intuitively, the complexity of parameterization of the Möbius bundle can be seen using horizontal transport. The illustration provided in Fig. 10.7 is helpful in describing this phenomenon. Assume that the base is the horizontal unit circle, and initially a point u on a vertical fiber is located at height 1/3, as measured from the base. Here it is assumed the fibers are 1 unit long, and horizontal angles are measured in degrees counterclockwise. The coordinates of u initially are (0, 1/3). Although the fibers are of the same length, they no longer stay vertical due to the flip as the angle changes. However, all measurements are made along the fiber from the base. If point u is transported by a change of angle of 360 degrees, point u is now at height 2/3 and located at point w. The trace for the movement of point u is illustrated in the figure.

A solid line with arrows is used when the sides of the Möbius strip can be observed in the front view. Broken lines with arrows are used for portraying hidden views. By rotating another 360 degrees, the point u does return to its original position. Equivalently, let w have initial coordinates (0, 2/3), and restrict the angle to stay between zero and 360 degrees. The trace is given again as w is transported. This is depicted in the diagram this time using solid dots for observable traces and nonsolid circles for hidden traces. See Fig. 10.7. Note that the traces of the points u and w must intersect at some point. It follows

that the parameterization is not global. The point where the paths cross the parameterization is not well defined.

Among the largest differences between the cylinder and the Möbius fiber bundles are the structure groups that are employed for construction. Transition functions within these groups are smooth and determine how the fibers are held together. These functions are used to smoothly change the parameterization when needed. For the Möbius strip, the transition functions are used to separate two possible parameterizations. Intuitively, a change in parameters occurs at the point where the paths intersect. All in all, these algebraic groups characterize the fiber bundle and are described further in the subsequent sections.

In more detail, a fiber bundle is comprised of three sets: the total space E, the base space B, and the fiber F. As previously seen in simple cases, $E = B \times F$, that is, E is just the product space. In this case, B is called the trivial bundle. Generally, there is a local continuous, onto projection $p: E \to B$. This map p is also an open map because projections of products are open maps. Moreover, this map is also known as the submersion of the bundle (Phillips, 1967). In more complex spaces, the map is involved; however, locally, it still acts like a projection, but it usually is not 1−1. Similar to short exact sequences, fiber bundles are described by:

$$F \to E \xrightarrow{p} B$$

For any point x in B, $p^{-1}(x)$ must be diffeomorphic to the fiber F. In general, local trivialization occurs for any u in E; this begins with the projection $p(u) = x$, where u is an element of the fiber above x. A fiber bundle is a set of trivializations that cover the base manifold. It is possible to use a map, $h(u) = (x, f)$, where for any open set U in B, $h: p^{-1}(U) \to U \times F$. Let h′ be another trivialization: $h'(u) = (x, f')$ where $h': p^{-1}(U) \to U \times F$. When these two trivializations overlap, h and h′ are related to a transition function based on x. An instance of this is illustrated in Fig. 10.9. For the overlap, the structure group G for the fiber bundle has elements g, depending on x, such that $f' = g\,f$. This is a smooth left action of G on F. Transition functions map fibers into fibers and are diffeomorphisms; they are in the structure group. This group is a subset of all diffeomorphisms on F. Recall that a trivial fiber bundle is a fiber bundle where $E = B \times F$ everywhere. Only one bundle chart is needed in this case.

10.5 Sections in a fiber bundle

A section of a fiber bundle is an identity-type mapping from B to E. The section includes points in F that are above point x in the base B, The projection p when applied to points in the section restores the original base point x.

Example 10.6:

Refer to the Möbius fiber bundle given above and in particular to Fig. 10.7. The entire continuous path in the transport for u in $[0, 2\pi)$ is a section. In this case, it is assumed there are no gaps in the path so that the projection of every point in the path restores the entire base S^1. In the figure, the start position is illustrated at the far end of the strip, and the ending point is approached in the limit on top of the back of the strip.#

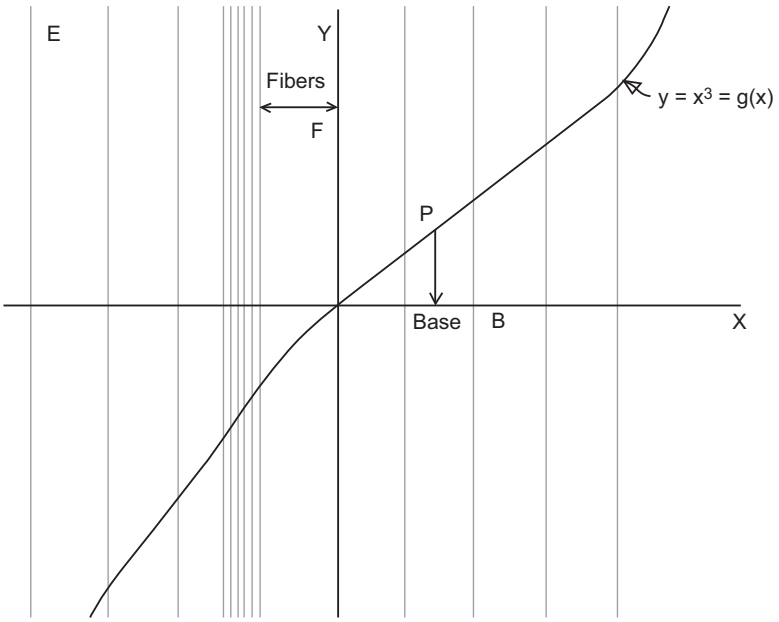


FIGURE 10.8 Sections in a fiber bundle.

Example 10.7:

This is a simpler example of a trivial fiber bundle when compared to the Möbius fiber bundle. Consider R^2 as the total space E. Let the base B be just the line R^1, the x axis. Also let the fiber be $F = R^1$ placed orthogonal to the original line B, also parallel to each other, and constituting a plane. This is a trivial bundle, that is, E = B x F. Intuitively think of the base as the x-axis, and the fibers are strands along and all parallel to the y axis. Moreover, if g: B→F, where $g(x) = x^3$, then g is a section. Accordingly, the projection map p(g)(x) provides all points on B. This is illustrated in Fig. 10.8.#

10.6 Line and vector bundles

Bundles in general consist of sets of trivializations covering the base manifold within a fiber bundle. Trivializations create bundles utilizing transition functions. Line bundles or vector bundles of rank one have fibers that are one-dimensional complex or real vector spaces. Additionally, the structure group acts linearly on the vector space. Rigorously, they are defined by letting L and B both be smooth manifolds with a smooth onto projection p: L→B. For every x in B, the fiber F of L over x, that is, $p^{-1}(\{x\})$, is a complex vector space of dimension 1. As specified earlier, there is a trivialization for every x. There is an open neighborhood U of x such that the diffeomorphism h: $p^{-1}(U) \rightarrow U$ x C. Furthermore, for the projections p1: U x C→U, and p2: U x C→C, first let p(u) = p1(h(u)) for all x in U.

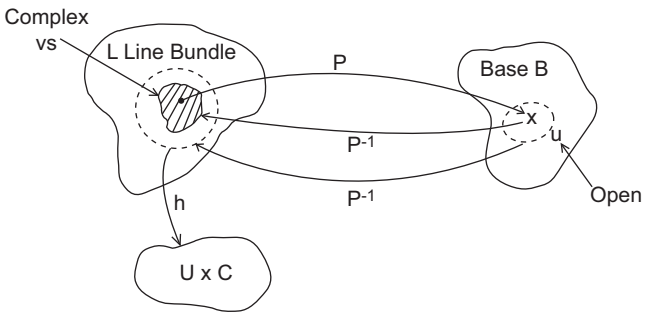


FIGURE 10.9 Trivialization in a line bundle.

In this case, p2(h(u)) is a vector space isomorphism of $p^{-1}(\{x\})$ with C. Fig. 10.9 illustrates some of these concepts.

Vector bundles have fibers that are real or complex n-dimensional vector spaces. Additionally, the structure group acts linearly on the vector space. Transition functions also act with n-dimensional operations on the fibers. These functions belong to subgroups within GL(n, R) or GL(n, C); both groups are general linear groups and are described in detail in the next chapter.

10.7 Analytic vector bundles

Let B be a complex manifold over C. An analytic vector bundle of rank n is a holomorphic map p: E→B, such that for any point x in B the preimage $Ex = p^{-1}(x)$ has an n-dimensional complex vector space structure. Here, Ex is a fiber. Additionally, the map p is locally trivial. This means that for any point x in the base B there exists an open neighborhood Uj, which contains x along with a bi-holomorphic map fj: $p^{-1}(Uj) \rightarrow Uj \times C^n$, where the projection p1: $Uj \times C^n = Uj = p(p^{-1}(Uj))$. Also, fj is an isomorphic mapping from Ex onto $\{x\} \times C^n$ for every x in Uj, also called a trivialization of E over U. As mentioned previously, the general linear group is involved. For any pair of trivializations fi and gj, the mapping hij: Ui intersects Vj→GL(n, C) where $hij(x) = fi(gj^{-1})$, but gj has a restricted domain of $\{x\} \times C^n$. This mapping is holomorphic and called transition functions for E relative to the trivializations fi and gj. The transition functions satisfy the identities:

1. Inverse: hij(x) hji(x) = I for all x in Ui intersect Uj.
2. Co-cycle Condition: hij(x) hjk(x) hki(x) = I for all x in the intersection of Ui, Uj, and Uk.

Vector bundles E can be created using transition functions hij: Ui intersect Uj→GL(n, C) where {Uk} is an open cover of B. This employs the gluing construction using Ui X C^n for all I, thus obtaining E = the disjoint union of $(Ui \times C^n)/\sim$. Here, ∼ refers to the equivalence class (x, v)∼(x, hij(x)(v)) for all x in Ui intersect Uj and v in C^n.

Example 10.8:

The trivial analytic vector bundle of rank n uses the projection map p1: $B \times C^n \rightarrow B$. For the vector bundle E over B, let hij be the transition functions and {Uj} an open cover of B.

The dual bundle is E* defined by transition functions $gij(x) = hij^{-1}(x)$, for all x in Ui intersect Uj. For an open set U in B, a holomorphic section s over U is s: U→E, and p(s(U)) = U. A global section of E is a holomorphic section where U = B. In this case, all such global sections form a vector space, denoted by Ho(B, E) (Chern, 1979).#

10.8 Elliptic curves over C

Elliptic curves are defined over several distinct fields. These carrier sets include the reals, the rationals, and Galois fields, as well as the current application, the complex field. Appendix A3 provides an introduction to the MSA involving the additive group structure for elliptic curves involving the real field. Later chapters will utilize elliptic curves over Galois fields for encryption purposes.

For an elliptic curve E, over C there always exists a lattice L, a subset of C that is unique up to scaling or homothety. Homothety preserves dilations and contractions about a center. For every lattice, there is an elliptic curve. More specifically, the elliptic curve is the quotient space of C by a lattice. Going the other way, any quotient space of C by a lattice is an elliptic curve. Additionally, there is a complex and analytic isomorphism g, g: C/L→E (Silverman, 2009). This is the set of points on the elliptic curve in the complex plane. It is an isomorphism of Lie groups. The uniformization theorem provides the connection between elliptic curves over C and lattices.

Example 10.9:

The elliptic curve is the cubic, $y^2 = x^3 + ax + b$, where a and b are now complex numbers. Also, $4a^3 + 27b^2$ is not zero, making the curve smooth. Let E = {(x, y), where x and y are in C and are points on the elliptic curve} union the point at infinity, ∞. In projective space, ∞ = (0: 1: 0) (Coxeter, 1989).#

A lattice L is a discrete Abelian additive group of C. It consists of the points that intersect the somewhat horizontal parallel lines with the somewhat vertical parallel lines. See Fig. 10.10.

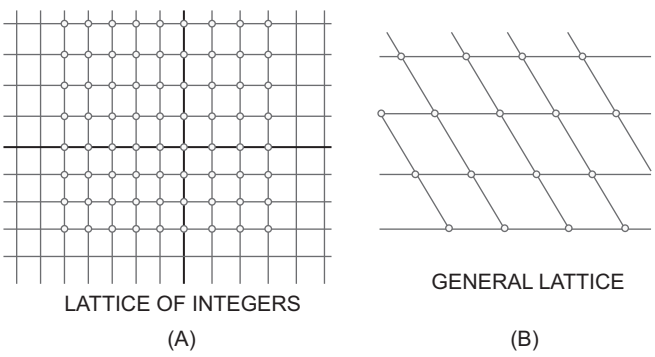


FIGURE 10.10 Two types of lattices. (A) Square pattern lattice, (B) more general lattice.

Example 10.10:

The simplest lattice is L = Z[i] illustrated in Fig. 10.10A. It is defined as L = {v = n + mi, such that n and m are in Z}. It follows that under usual complex field operations, this is an additive Abelian group. A more typical lattice is illustrated in Fig. 10.10B. In general, the parallelograms in a lattice are all of the same dimension and equally spaced.#

An isogeny h, of elliptic curves E1 and E2 is a nonzero holomorphic group homomorphism such that h: E1→E2, where g(0) = 0. When E1 = E2 = E, then the isogeny is called an endomorphism from E into itself, and it is abbreviated as End(E). An endomorphism ring on E is an endomorphism on E with addition being defined by the group structure point-wise, and multiplication is defined as function composition. This is denoted by End (E). An in-depth treatment can be found in Silverman (1994).

Example 10.11:

If for every p in E, g(p) = n p, where n is some point in Z, then this is an endomorphism on E. This is called the multiply by n map. Consider, the elliptic curve E_1, $y^2 = x^3 - x$, and the elliptic curve E_2: $y^2 = x^3 + 4x$ Then, the mapping g: $E_1 \to E_2$, where $(x, y) \to (y^2/x^2, -y(1 + x^2)/x^2)$, and h: $E_2 \to E_1$, where $(x, y) \to (y^2/(4x^2), -y(4 + x^2)/(8x^2))$. Next, the coordinate image under g will be substituted into the elliptic curve E_2. That is, $[-y(1 + x^2)/x^2]^2 = [y^2/x^2]^3 + 4[y^2/x^2]$, this quantity also can be written as follows: $[y^2(1 + 2x^2 + x^4)x^2] = y^6 + 4y^2x^4$, when x differs from zero. Writing this again gives the result: $y^4 = x^6 - 2x^4 + x^2 = (x^3 - x)^2$. One branch yields the elliptic curve E_1. A similar conclusion would happen if the image of h was substituted into the elliptic curve E_1. The crux of all this is that h(g)(.) is a multiply by two map from E_1 into itself. Likewise, g(h)(.) is a multiply by two map from E_2 into itself (Lin, 2021).

Example 10.12:

The map z mod(Z[i]→iz mod(Z[i]) is an endomorphism for $y^2 = x^3 - x$. Since g: $(x, y) \to (-x, iy)$, substituting into the curve yields $(iy)^2 = (-x)^3 - (-x)$, so $y^2 = x^3 - x$, the same curve. Notice that g^4 is the identity map. However, g is not a multiply by n map. Because the order of points (x, y) for the elliptic curve and for g, it is (−x, iy), they differ. To see that g^4 provides the identity map, apply the substitution $(x, y) \to (-x, iy)$ over and over four times. Thus, g^4: $(x, y) \to (-x, iy) \to (x, -y) \to (-x, -iy) \to (x, y)$.#

An elliptic curve E has complex multiplication (CM), which means that Z is not the only endomorphism in End(E). The previous example is an illustration of CM. When E does not have CM, then the endomorphism is an isomorphism, that is, g: E→E, g(p) = n p, for fixed n in Z.

10.9 The quaternions

The quaternions can be described in various ways. They form a four-dimensional vector space over the reals. Additionally, the quaternions form a skew field or a division algebra.

It is also a normed, unital, noncommutative, associated algebra over the reals. Similar to the complex number system, there is a conjugate operation acting in an analogous manner. Indeed, for the quaternion $q = a + bi + cj + dk$, the conjugate of q is denoted by q* and it equals $a - bi - cj - dk$. Moreover, conjugation is an involution, that is, $(q^*)^* = q$ and for two quaternions $(p\ q)^* = q^*\ p^*$.

A quaternion p is unital when $|p| = 1$. The most important unital quaternions are the single unit quaternions: 1, i, j, and k. V-ONE = 1. The others obey the multiplication identities: $ij = k$, $jk = i$, $ki = j$, $ji = -k$, $kj = -i$, $ik = -j$, $ii = -1$, $jj = -1$, and $kk = -1$. Unlike conjugation in the complex field, conjugation for quaternions can be performed utilizing the defining operations as given in the MSA for fields, along with the equational identities. In this case, the quantity q to be conjugated equals minus one-half times the sum involving all single unital quaternions. These unital quaternions multiply q on both sides, resulting in the formula $q^* = -(1/2)\ [1q1 + iqi + jqj + kqk]$.

Example 10.13:

Given the quaternion $q = 2 - j + 4k$. The conjugate quaternion is $q^* = 2 + j - 4k$. However, utilizing operations within the skew field specification, the conjugate can again be found. Thus, in a lengthy calculation, $q^* = -(1/2)\ [(2 - j + 4k) + i(2 - j + 4k)i + j\ (2 - j + 4k)j + k(2 - j + 4k)k]$. Now multiply by the left single quaternion, $q^* = -(1/2)\ [(2 - j + 4k) + (2i - k - 4j)i + (2j + 1 + 4i)j + (2k + i - 4)k]$. Next multiply by the right single quaternion, $q^* = -(1/2)\ [(2 - j + 4k) + (-2 - j + 4k) + (-2 + j + 4k) + (-2 - j - 4k)]$. Finally, add everything together gives $q^* = -(1/2)\ [-4 - 2j + 8k] = 2 + j - 4k$.#

The norm of p is $\|p\|$; it is usually written as $|p|$, and it when squared equals $a^2 + b^2 + c^2 + d^2$. Being a division algebra, it is such that every nonzero element has an inverse on the left or on the right. Thus, for p nonzero it follows that either pq^{-1} or $q^{-1}p = 1$, where $q^{-1} = q^*/\|q\|^2 = (a - bi - cj - dk)/(a^2 + b^2 + c^2 + d^2)$.

It will be seen that the quaternions have a group structure on the 3 sphere, resulting in isomorphisms with the special unitary group SU(2), as well as similarities with the special orthogonal group SO(3). These groups were previously mentioned in connection with the Bloch sphere. Unit quaternions are isomorphic to the special unitary group consisting of unitary matrices with determinant one. Integer or half-integer quaternions are also utilized in describing symmetries in regular polyhedra.

Every quaternion p can be written as a real part po and a three-dimensional vector part v. Here, $v = bi + cj + dk$. A unit quaternion can be written as $p = \cos(t) + v \sin(t)$ where $|v| = 1$ and t is in $[0, 2\pi)$. Similar to the complex field situation, the scalar part of $p = (1/2)(p + p^*)$ and the vector part of $p = (1/2)(p - p^*)$. Let $q = qo + w$, where w is the vector part. Then, the product of the two quaternions $p = po + v$ times q equals $(po\ qo - <v, w>) + (po\ w + qo\ v + v \times w)$. Here, the first set of parenthesis is scalar-valued with $< v, w >$ being the usual dot product in R^3. The second set of parenthesis contains the vector part of the product p q. The quantity $v \times w$ is the cross product in R^3. The final representation of quaternion q involves the complex numbers c_1 and c_2. Write quaternion $q = c_1 + c_2\ j$, where $c_1 = a + bi$ and $c_2 = c + di$, then $q = a + bi + cj + dk$ as before.

Example 10.14:

The quaternions Q, being an algebra among other things, can be made into a Lie algebra LQ, where the Lie bracket is utilized. That is, use for p, q in Q the commutator: [p q − q p]. In particular, the Lie algebra of derivations D is the Lie algebra R^3. This is shown below where the commutator is related to the cross product in R^3, [p q − q p] = 2 p x q. Note that the center of LQ is the reals R, because for any real, say po and any quaternion q, it follows that [p, q0] = qop − qop equals zero.#

Scalars commute with quaternions. It is said that the real quaternions form the center of the quaternion algebra. So R is also the kernel. Next, form the factor Lie algebra LQ/R. Take the Lie bracket [p, q] using v and w as the vector parts of p and q. So p q − q p = (po qo − < v, w>) + (po w + qo v + v x w) − (qo po − < v, w>) −(qo v + po w + w x v). Note that the real part equals zero as well as all vector parts except for the cross product terms. The result is 2 (v x w) = v x w − w x v, as mentioned in the previous example.

10.10 Hopf fibrations

There are four different Hopf fibrations all solely involving spheres. The fiber F the base B, as well as the total space E, all involve spheres. These four fibrations are denoted using the schema:

$$F \to E \to B$$

$$S^0 \to S^1 \to S^1$$

$$S^1 \to S^3 \to S^2$$

$$S^3 \to S^7 \to S^4$$

$$S^7 \to S^{15} \to S^8$$

The first fibration mentioned earlier involves a point S^0, along with the unit circle S^1, and is illustrated below in the example. The second fibration makes use of complex numbers in deriving results for relationships between the fiber, the base, and the overall structure. The procedure is developed below while relating the fibration to the Bloch sphere. See Section 10.11. The third fibration forms the contents of Section 10.12. In this case, quaternions are employed in showing Hilbert space relationships between two qubits and their possible entanglement. Finally, the last fibration employs octonions and is not explained herein.

Example 10.15:

Pairs of antipodal points on a circle are mapped to a single point on a new circle for the first fibration. The process is employed in two steps and is illustrated in Fig. 10.11. The original points along with antipodal points are located in Fig. 10.11A. Pairs of points are labeled j and j′, j = 1, 2, 3, 4, 5. The first operation is illustrated in Fig. 10.11B, to the

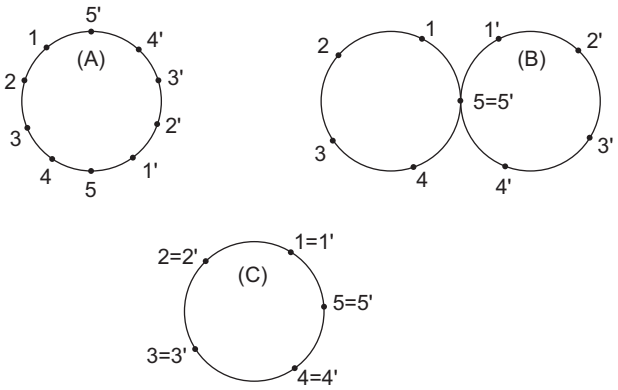


FIGURE 10.11 Hopf fibration $S^0 \rightarrow S^1 \rightarrow S^1$. (A) Original circle; (B) Twist applied to circle; (C) Folding operation.

right. Here, the original circle is twisted into a figure eight. In this diagram, antipodal points are still identifiable, except for point $5 = 5'$. Secondly, folding the left half of figure eight onto the right half yields the end result. It is a circle in which antipodal points are mapped into single points on a new circle. This can be seen in Fig. 10.11C. In the lower figure. The resulting space is the quotient space S^1.

10.11 Hopf fibration with bloch sphere S^2, the one-qubit base

This presentation is entirely based on Mosseri and Dandoloff (2001). It involves the mappings, $S^1 \rightarrow S^3 \rightarrow S^2$. Specifically, the Hopf fibration with base $B = S^2$ is a most important example of a nontrivial fibration. In this structure, $E = S^3$ is fibered with great circles, $F = S^1$. Due to the nontriviality, E is not equal to B x F everywhere, except as usual they are equal locally. In terms of qubits, the fiber is identified with the uncertain global phase, ϕ of a general qubit, $e^{i\phi} \; |q>$. The base can be thought to be the Bloch sphere with pure qubits occupying its surface, and mixed qubits lie within. As mentioned earlier, the fibration will be realized using complex values, u and v in C where $u = x + iy$ and $v = z + it$, with x, y, z, and t real valued. Since S^3 is defined as $x^2 + y^2 + z^2 + t^2 = 1$, it can be defined using u and v by $|u|^2 + |v|^2 = 1$. The Hopf fibration in this case will be described using the composition of two mappings H followed by K. The first map sends S^3 into the extended plane R^2 union $\{\infty\}$. The second map sends this extended plane into S^2 using an inverse stereographic projection. Specifically, these maps are:

$$H(u, v) = \left(u\, v^{-1}\right)^*$$

$$K\left(\left(u\, v^{-1}\right)^*\right) = (X, Y, Z), X^2 + Y^2 + Z^2 = 1$$

The great circle $\left(u\, e^{i\phi}, v\, e^{i\phi}\right)$ in S^3, parametrized by ϕ, is mapped into the single point, that is, $H\left(u\, e^{i\phi}, v\, e^{i\phi}\right) = \left(u\, v^{-1}\right)^*$.

Substituting into the first mapping H (u, v) = $(u\, v^{-1})^* = (x - iy)/(z - it) = [(x\, z + yt) + i(x\, t - yz)]/(z^2 + t^2)$.

10.12 Hopf fibration with sphere S^4, the two-qubit base

This presentation is entirely based on Mosseri and Dandoloff (2001). The Hopf fibration involving the sphere $E = S^7$, with fibers, $F = S^3$, and base S^4, is developed similar to $S^1 \to S^3 \to S^2$. In place of complex numbers, quaternion numbers are used. The whole treatment parallels the fibration given in Section 10.11. In this case, the fibration: $S^3 \to S^7 \to S^4$ will be illustrated. The procedure follows that of the referenced paper. For complex numbers, $u_p = x_p + iy_p$, and $v_p = z_p + it_p$, with x_p, y_p, z_p, and t_p real valued, $p = 1$ or 2; the quaternion numbers $q_p = u_p + v_p j$ will be used for the Hopf fibration. Also, $|q_1|^2 + |q_2|^2 = 1$ and so $x_1^2 + y_1^2 + z_1^2 + t_1^2 + x_2^2 + y_2^2 + z_2^2 + t_2^2 = 1$, showing that the pair of quaternions q_1 and q_2 represents a state lying on the surface of S^7.

The Hopf fibration in this case will again be described using the composition of two mappings H followed by K. The first map sends S^7 into the extended surface R^4 with $\{\infty\}$. The second map sends this extended surface into S^4 using an inverse stereographic projection. As previously mentioned, the method parallels the complex situation almost 100%. The big differences are that the dimension in this case is larger, and more importantly, the quaternions form a noncommutative field-type structure. The two mappings are as follows:

$$H(q_1, q_2) = (q_1 q_2^{-1})^*$$

$$K((q_1\, q_2^{-1})^*) = (X, Y, Z, U, V), X^2 + Y^2 + Z^2 + U^2 + V^2 = 1$$

Care has to be taken in expanding the first map H, because in general, quaternions do not commute. Beginning with the first mapping: $H(q_1, q_2) = (q_1 q_2^{-1})^* = (q_2^{-1})^* q_1^*$. Taking q_2^{-1} gives $q_2^{-1} = q_2^* / \|q_2\|^2 = (x_2 - y_2 i - z_2 j - t_2 k)/(x_2^2 + y_2^2 + z_2^2 + t_2^2)$. As a consequence, $(q_2^{-1})^* = (x_2 + y_2 i + z_2 j + t_2 k)/(x_2^2 + y_2^2 + z_2^2 + t_2^2)$. So, $H(q_1, q_2) = (q_2^{-1})^* q_1^* = (x_2 + y_2 i + z_2 j + t_2 k)(x_1 - y_1 i - z_1 j - t_1 k)/(x_2^2 + y_2^2 + z_2^2 + t_2^2)$. Next, let $D = (x_2^2 + y_2^2 + z_2^2 + t_2^2)$, and using the complex quantities, $u_p = x_p + iy_p$, and $v_p = z_p + it_p$, with $q_p = u_p + v_p j$, for $p = 1, 2$. This yields the expression: $H(q_1, q_2) = [(u_2 + v_2 j)(u_1^* - v_1 j)]/D = [(u_2 u_1^* + v_2 v_1) + (v_2 u_1^* - u_2 v_1)j]/D = [(u_1^* u_2 + v_1 v_2) + (u_1^* v_2 - v_1 u_2)j]/D$. In the referenced paper (Mosseri and Dandoloff, 2001), the quantity u_1^* appears without the conjugation, and the term multiplying j is set to zero. There, they interpret or relate the pure complex result to the criteria for simple tensors, or in their case, nonentangled states.

References

Chern, S., 1979. Complex Manifolds Without Potential Theory. Springer.
Coxeter, H., 1989. Introduction to Geometry. Wiley.
Dugundji, J., 1975. 978-0697-006889-7 Topology. Allyn and Bacon.
Halmos, R., 1974. 0-387-90093-4 Finite Dimensional Vector Spaces. Springer.
Lee, J., 2006. 978-0387-95448-6 Introduction to Smooth Manifolds. Springer-Verlag.
Lin, A., 2021. Complex Multiplication and Elliptic Curves. Stanford.edu.

Mosseri, R., Dandoloff, R., 2001. Geometry of entangled states, Bloch spheres, and Hopf fibrations. J. Phys. A: Math. Gen. .
Munkres, J., 1999. 0-13-181629-2 Topology. Prentice Hall.
Phillips, A., 1967. Submersions of open manifolds. Topology 6.
Rudin, W., 1966. 0-07-054234-1 Real and Complex Analysis. McGraw-Hill.
Silverman, J., 1994. Advanced Topics in The Arithmetic of Elliptic Curves. Springer.
Silverman, J., 2009. The Arithmetic of Elliptic Curves. Springer.

This page intentionally left blank

CHAPTER

11

Lie algebras and Lie groups

11.1 Algebraic structure

A Lie algebra is a nonunital often, nonassociative algebra. Accordingly, it has a binary operation similar to multiplication; however, it is different from all those binary operations previously mentioned. The multiplication is not a functional calculus type, not convolution, nor actually operator composition. It is called the Lie bracket and is an alternating bilinear map that must satisfy several equational constraints beyond those for a vector space or an algebra. These constraints are rigorously specified below in the MSA global view. As in the concept of Hilbert space, there exists an unlimited number of distinct Lie algebras. Each algebra has its own carrier sets corresponding to the sorts, as well as operators described from the signature sets.

Like a Hilbert space, Lie algebras have topological as well as other mathematical structures besides the algebraic structure. Moreover, in finite dimensions, every Lie algebra can be associated with one or more Lie groups. In general, a Lie algebra over vector fields is a Lie algebra of a diffeomorphism group for a manifold. A diffeomorphism is an isomorphism of smooth manifolds. It maps one manifold to another, and it and its inverse are differentiable. In any case, there exists a strong connection between a Lie algebra and group structures. This correspondence exploits Lie algebra to better understand and categorize the groups involved. Much of this correspondence will be illustrated later.

11.2 MSA view of a Lie algebra

To begin, a high-level description of a Lie algebra in the MSA utilizes two sorts: SCALAR and VECTOR. The carrier sets corresponding to SCALAR are usually either complex or real numbers, and the signature sets are those of a field. Accordingly, the complex or real field results in a lower view in the MSA. When Lie groups are considered, the sort SCALAR will not only include complex and real numbers; it will also include quaternion numbers. Consequently, the signature sets and equational identities will be those of a skew field in the latter case. For a Lie algebra, as in an associative or nonassociative algebra, there exists a binary function BINE such that:

$$\text{BINE} : \text{VECTOR} \times \text{VECTOR} \rightarrow \text{VECTOR}$$

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00012-6

Besides all the equational identities for vector space and an algebra, the operational name BINE must satisfy the following four identities. First, replace the sorts and operational names for:

VECTOR by u, v, w.
SCALAR by a, b.
BINE by [.,.], this is often called a Lie bracket.
V-ADD by +.
S-MULT by ·.

The equational identities are the following:

1) Bilinearity: $[a \cdot v + b \cdot u, w] = a \cdot [v, w] + b \cdot [u, w]$.
2) Bilinearity: $[w, a \cdot v + b \cdot u] = a \cdot [w, v] + b \cdot [w, u]$.
3) Anticommutative: $[u, v] = -[v, u]$.
4) Jacobi identity: $[u, [v, w]] + [v, [w, u]] + [w, [u, v]] = 0$.

Note that by using (3) it follows that $[u, u] = -[u, u]$ implies that $[u, u] = 0$. Also, note that the two distributive laws for an algebra are upheld by using the two bilinear relations (1) and (2). Similarly, the multiplicative homogeneity condition for an algebra also holds. It holds again by using the two bilinear conditions. The possible nonassociativity means that [x, [y, z]] need not equal [[x, y], z]. Fig. 11.1 illustrates the polyadic graph for a Lie algebra. Not shown in this diagram are all the operational names solely associated with sort SCALAR.

11.3 Dimension of a Lie algebra

The dimension of a Lie algebra is the dimension of its underlying vector space over the reals or complex numbers. A subalgebra M, of a Lie algebra L, is a subspace where the Lie bracket is closed. Thus, if for all v and w in M, there exists a u in M such that $u = [v, w]$; then M is a sub-Lie algebra (Jacobson, 1979).

Example 11.1:

Among the simplest examples of a Lie algebra is where sort VECTOR has the carrier set of all real-valued three-dimensional vectors. In this case, the cross product operation x acts

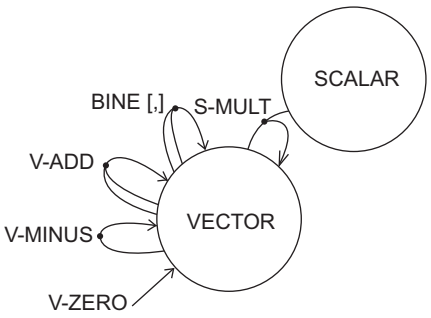


FIGURE 11.1 Lie algebra MSA graph.

as the Lie bracket in R^3. The standard i, j, and k coordinate system is most often employed in this application with i x j = k, j x k = i, k x i = j. Whenever the operands in the product are transposed, the result is negated. This establishes criteria (3) anticommutative. Moreover, the cross product of any vector with itself is zero. The Jacobi identity follows immediately since (i x (j x k)) = i x i = 0. As an example, say that the vectors are v = 1i + 2j + 3k and w = −1i + 4j−2k; then v x w can be found by taking the determinant of:

$$\begin{vmatrix} i & j & k \\ 1 & 2 & 3 \\ -1 & 4 & -2 \end{vmatrix} = \begin{vmatrix} i & j & k & i & j \\ 1 & 2 & 3 & 1 & 2 \\ -1 & 4 & -2 & -1 & 4 \end{vmatrix}$$

So the result is v x w = −4i−3j + 4k + 2k −12i + 2j = −16i − j + 6k.#

Basis-independent criteria can be employed showing R^3 is a Lie algebra over the reals. Here, we use the fact that R^3 is a real-valued Hilbert space H. For u, v, and w in H, use the real-valued dot product as the inner product <v, w>. In this case, by direct calculation, it follows that the formula holds: u x (v x w) = v <w, u>−w <u, v>; then substituting into the left side of the Jacobi identity, it becomes u x (v x w) + w x (u x v) + v x (w x u) = v <w, u>−w <u, v> + u <v, w>−v <w, u> +w <u, v>−u <v, w> = 0.#

Example 11.2:

An associative algebra A over the real or complex field becomes a Lie algebra by introducing the Lie bracket. In this case, it is called the commutator. The commutator is defined for v and w in A and is given by [v, w] = v•w−w•v. The multiplication v•w and w•v are the inherited multiplication in the not necessarily commutative associative algebra A. Accordingly, now there exist two types of multiplication: one for the associative algebra and the other for the Lie algebra. #

In finite dimensions, the Lie algebra of n by n matrices is described with the Lie bracket being the commutator. These Lie algebras are denoted by gl(n,C) or gl(n, R) depending on whether the field is complex or real. These are the general linear Lie algebras corresponding to the general linear groups of invertible matrices. So, the determinants of these matrices are nonzero.

There are numerous properties relating Lie groups to specific Lie algebras. In particular, Lie groups have an identity, usually a matrix related to the zero vector in the Lie algebra. Additionally, the Lie group is locally invertible; there are neighborhoods about the identities allowing continuous maps between these open sets. It will be seen that the exponential is a mapping from the Lie algebra to the Lie group. Conversely, there is a linearization or tangent operation from the Lie group to the Lie algebra. Essentially, the tangent space becomes the Lie algebra. These are methods for associating Lie algebras with certain Lie groups and associating Lie groups with a unique Lie algebra.

Example 11.3:

The Campbell-Baker-Hausdorff (CBH) formula will show the large use of Lie algebra in determining local behavior for elements near the identity of Lie groups. This formula is provided as an infinite series that need not converge, and the presentation is formal. The group multiplication of the exponential group is determined by the bracket of the Lie

algebra in this case. The formula involves the Lie bracket as well as the power series expansions for exp and log. The CBH formula involves $\log((e^a)(e^b))$, where a and b are elements from a Lie algebra, which do not commute. Use for log (1 + z), the Taylor expansion about the origin, this gives $\log(1+z) = z - z^2/2 + z^3/3 - z^4/4 + \ldots$. This series converges for $|z| < 1$, and it is used for only a small neighborhood of the origin. Also, let, $e^z = 1 + z + z^2/2 + \ldots$, which converges everywhere in C. This is the usual power series expansion for the exponential e^z. Next, the power series expansion for the logarithm of the product of two exponential will be found. The terms will be illustrated one by one, in order of their polynomial degree.

For $\log((e^a)(e^b))$, there is no constant term, but:

There are two linear terms and when added together are a + b.
There are second-order terms given as $ab/2 - ba/2$; these terms can be described by using the Lie bracket as 1/2 [a, b].
There are third-order terms that are $1/12\ (a^2 b + a s^2 + b^2 a + b a^2 - 2 a b a - 2 b a b)$.
As previously seen from the lower order terms, this expression can also be written using the Lie bracket. In this case, it equals,1/12 ([a, [a, b]] + [b, [b, a]]).
There are fourth-order terms, and they are 1/24 [a, [b, [a, b]]]. . . .#

The CBH formula shows that all terms of the expansion can be written exclusively using Lie brackets. Also, of interest, Dynkin's formula enables calculation for a large number of commutator terms within a partial sum series for the CBH formula (Hall, 2015).

Example 11.4:

Consider the unit circle in the complex plane, $S^1 = U(1) = \{z, \text{such that } z \text{ is in C and } |z| = 1\}$, then $z = e^{it}$. The Lie bracket [it, is] = 0, for s and t real. Accordingly by CBH, just using the linear terms, it follows that $e^{it}e^{is} = e^{i(s+t)}$. All other terms in the CBH formula are zero. Taking the tangent line at the point (1, 0) on the unit circle, here the associated Lie algebra is parallel to the tangent line at (1, 0). To see this, first note that at time t = 0 the point on the circle path is (1, 0). Next taking the derivative of z with respect to the parameter t is $dz/dt = i e^{it}$, and evaluating at t = 0 gives i. So the direction in the complex plane is pointing upward along the upper part of the ordinate. Translating this vector to the point (1, 0) again gives a tangent vector pointing upward. #

Another identity related to the CBH formula is the Lie–Trotter formula. In this case, it involves exponentials raised to n by n real or complex matrix powers. In short, let the matrices be A and B; then $e^{(A+B)} = \lim_{k\to\infty} [e^{A/k} e^{B/k}]^{1/k}$. When a finite value of k is used, an approximation is obtained. Better approximations are obtained using the Suzuki-Trotter product formula: $e^{(A+B)}$, which is about equal to $e^{B/2} e^{A} e^{B/2}$. Higher order approximations can be found in Berry (2006).

11.4 Ideals in a Lie algebra

An ideal in a Lie algebra L is a subalgebra M having special properties. It is such that it is not only true that for all v, w in M, there is a u in M where [v, w] = u, but also for every

z in L and for every w in M, there is a v in M such that [z, w] = v. All ideals in L are two sided. Also, L itself is an ideal as well as the kernel of L. The factor or quotient algebra is L / M; it is an algebra of cosets of the form: L + M = [v + M, w + M] = [v, w] + M, for all v and w in L. The factor algebra is a Lie algebra using the Lie bracket. A Lie algebra is said to be simple when the only ideals in L are 0 and L whenever [L, L] is not zero, that is, it is not commutative. Examples of a nonsimple Lie algebra are the general linear algebras, gl(3,C) or gl(3,R). These algebras are often referred to as gl(3) for short. In any case, it is easy to see that all scalar matrices in gl(3) constitute a subalgebra. Moreover, matrices within this subspace have the property that for a and b scalars, the commutator of diagonal matrices [aI, bI] = 0.

Example 11.5:

Within g(3), a special subalgebra will be described. In this case, sort VECTOR consists of all three by three matrices that are upper triangular. The sort SCALAR can be the reals or the complex numbers. In any case, signature sets correspond to the usual operations for associative algebra over a field. This structure S is a Lie algebra using the Lie bracket: [A, X] = A.X−X.A, where A and X are upper triangular and A.X denotes the operation BINE (A, X) in the associative algebra of 3 by 3 matrices. An important fact is that the sub-Lie algebra of strictly upper triangular matrices forms an ideal of S. These types of matrices T only allow nonzeros above the main diagonal; consequently, all zeros appear on and below the main diagonal. They are also called the Heisenberg matrices. So, consider X in S and T in the ideal mentioned above; then, [X, T] results in another Heisenberg matrix. Given below are X, T, and the commutator X.T−T.X in that order.

$$\begin{array}{cccccccccc} |a & b & c| & |0 & x & y| & |0 & (a-d)x & (a-f)y-ex+bz & | \\ |0 & d & e| & |0 & 0 & z| & |0 & 0 & (d-f)z & | \\ |0 & 0 & f| & |0 & 0 & 0| & |0 & 0 & 0 & |. \end{array}$$

This shows that the strictly upper triangular matrices form an ideal among all upper triangular matrices within S.#

Example 11.6:

In this example, consider the carrier set for VECTOR to be all 2 by 2 real-valued matrices with trace equaling zero. In this case, SCALAR refers to the real-valued field, and the matrices are over the real field also. Let the operation BINE for the Lie bracket be the commutator. For this Lie algebra, a basis consists of the three matrices: u, v, and w given below, in their respective order:

$$\begin{array}{cccccc} |0 & 1| & |0 & 0| & |1 & 0| \\ |0 & 0| & |1 & 0| & |0 & -1| \end{array}$$

An arbitrary element x of the special linear Lie algebra, sl(2, R), is given by a matrix of form having trace zero given to the left, below. It is a linear combination of the three matrices u, v, and w above. The commutator will now be employed using the basis matrices mentioned earlier. Begin by forming the commutator of x with the first two basis

elements mentioned earlier. First, using u, the first basis element mentioned earlier gives y = [u, x] = (u.x − x.u). The resulting matrix is the matrix for y, which is illustrated in the center image. Lastly, using the commutator again, let z = [v, x] = (v.x − x.v), where x.v is the usual matrix product of a general element x, of sl(2, R), and a basis matrix v mentioned earlier. These matrices x, y, and z are, respectively,

$$\begin{vmatrix} a & c \\ b & -a \end{vmatrix} \quad \begin{vmatrix} b & -2a \\ 0 & -b \end{vmatrix} \quad \begin{vmatrix} -c & 0 \\ 2a & c \end{vmatrix}.$$

Arbitrary elements having the structure of x in sl(2, R) form a Lie algebra using this commutator. The set of all elements of structure y form a sub-Lie algebra of sl(2, R) and so do all elements of the form z. Moreover, structures such as these are said to be inner ideals of sl(2, R), because the Lie algebra sl(2, R) is simple; there are no actual ideals besides zero and sl(2, R) itself.#

In general, however, there are inner ideals that are not even subalgebras (Benkart, 1977).

The aforementioned example illustrating the existence of inner ideals in a simple Lie algebra also holds true in gl(2, C); all statements remain the same. Sort VECTOR in this case would be the set of all 2 by 2 complex-valued matrices of trace zero, and sort SCALAR would be the complex numbers.

Example 11.7:

For the Lie algebras gl(2, R), gl(2, C), sl(2, R), or sl(2, C), as in previous examples, a basis for the latter two Lie algebras consists of the matrices: u, v, and w. They are given below in their respective order:

$$\begin{vmatrix} 0 & 1 \\ 0 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & 0 \\ 1 & 0 \end{vmatrix} \quad \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}.$$

Notice that [w, u] = 2u, [w, v] = −2 v, and [u, v] = w. These relations involving the Lie brackets are satisfied for arbitrary k by k matrices, k = 2, 3, . . . For instance, let k = 3.

Call these three matrices U, V, and W. Let the nine entries of each matrix be addressed exactly as given next for U. The entries for U will be denoted by Uij where i is the row entry i = 1, 2, 3 and j is the column entry j = 1, 2, 3. Then for all the matrices, assume that all entries are equal to zero unless specified otherwise. Begin for U, let U12 = 2, and assume that U23 = 1. For V, let V21 = 1 and let V32 = 2. For W, let W11 = 2 and use W33 = −2. The following three-by-three matrices are in order, U, V, and W

$$\begin{vmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{vmatrix} \quad \begin{vmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{vmatrix}$$

Then, analogous to the two-by-two situation, it follows that [W, U] = 2U, [W, V] = −2V, [U, V] = W.

For k = 4, again the first two matrices X and Y are triangular upper diagonal and lower diagonal, respectively, and the third matrix Z is again diagonal. Using the same notation

as for k = 3, let all matrices have 0 entries except for the matrix entries: X12 = 3, X23 = 2, X34 = 1, Y21 = 1, Y32 = 2, Y43 = 3, Z11 = 3, Z22 = 1, Z33 = −1, and finally, Z44 = −3. If the Lie commutator brackets are used, then as before [Z, X | = 2 X, [Z, Y] = −2Y, and [X, Y] = Z.#

Generalizing the results in the previous example for n + 1 by n + 1 matrices, the Lie bracket relation again holds. Let A be the upper triangle diagonal, B the lower triangle diagonal, and C the diagonal. Again all matrix entries are zero except for A12 = n, A23 = n − 1, and Ak(k + 1) = n + 1 − k, k = 1, 2, . . ., n. Next, for B21 = 1, B(k + 1)k = k, k = 1, 2, . . ., n. Finally, for n even, then C11 = n, Ckk = n + 2 −2 k, k = 1, 2, . . ., n/2, and Cnn = −n, C(n − k + 2)(n − k + 2) = −n + 2k −2, k = 1, 2, . . ., n/2, and for the central entry, C(n/2 + 1)(n/2 + 1) = 0. Finally, for n odd, then C11 = n, Ckk = n + 2 −2 k, k = 1, 2, . . ., (n + 1)/2, and C(n + 1)(n + 1) = −n, C(n − k + 2)(n − k + 2) = −n+ 2k − 2, k = 1, 2, . . ., (n + 1)/2. Interesting descriptions of this can be found in Draper and Meulewaeter (2022).

The centralizer sub-Lie algebra of a set M is a set CM of elements that commute with M, that is, CM = {v in L is such that for all w in M, [v, w] = 0}. The center of L is the centralizer of L. A derivation D on the Lie algebra L is a linear mapping D: L → L, defined using the adjoint operation adu(v) = [u, v]. These derivations form a Lie subalgebra. Moreover, derivations satisfy Leibniz rule: D([u, v]) = [D(u), v] + [u, D(v)]. A Lie algebra homomorphism h, from one algebra to another, say h: Lie^1 into Lie^2, is a linear mapping such that h([u, v]) = [h(u), h(v)] for all u and v in Lie^1.

11.5 Representations and MSA of a Lie group of a Lie algebra

A representation of a Lie algebra A consists of a family of self-adjoint operators, skew self-adjoint operators, and symmetric or skew-symmetric operators. In the following, it will be assumed that the representation is a family F of self-adjoint matrices. For each generator, Xi in A, there corresponds a matrix Mi in F. The same commutation algebra holds [Xi, Xj] = [Mi, Mj]. An irreducible representation occurs whenever there are no trivial invariant subspaces of column vectors using any Mj. Two representations are equivalent; that is, M and N are equivalent meaning that there is a unitary matrix U such that N = U* M U (Hall, 2013).

Lie groups differ from discrete groups in that there are always some topological attributes associated with or defining the group. These include compactness and connectedness as well as separation axioms. Moreover, there are various manifolds, such as smooth, differentiable, or even holomorphic, also involved. As previously mentioned, in general these manifolds resemble Euclidean space on a small scale everywhere. Elements of the group are points in the manifold. Moreover, continuity plays a crucial role. As suggested early on, these groups are similar to the additive vector groups. However, Lie groups most of the time are of the multiplicative variety, but they do not have to be multiplicative; see Example 11.11. In any case, they are called continuous parameter groups. Their use is usually in modeling continuous symmetries, particularly rotational symmetries. Various types of Lie groups will be described, including real Lie groups, matrix Lie groups, and complex Lie groups.

The simplest Lie groups are the finite-dimensional real groups. In the MSA description, there exists one sort ELEMENT and it is the name for the members of the group. As in any group, the arity structure is (1, 1, 1). There exist three distinct signature sets, each containing a distinct arity operator name:

Zero-ary, IDENTITY,
Unary, INV: ELEMENT → ELEMENT
Binary, MULT: ELEMENT x ELEMENT → ELEMENT

However, unlike a discrete group, additional requirements prevail. ELEMENTS will have corresponding carrier sets being manifolds M, often Euclidean. Of most importance is that these manifolds enable calculus-type operations to be performed such as differential tangent space creation. In short, Lie group algebraic operations must be at least continuously differentiable. Derivative-type operations are usually conducted in a neighborhood of the most important element in the sort, the IDENTITY, I. For any element of the group, a mapping can be found transforming this nonunital element into the identity. Accordingly, two tangent spaces can be naturally identified resulting in a nonvanishing vector field on the group manifold. The existence of smooth vector fields on a manifold is termed parallelization.

Additionally, not only must there just exist mappings x and g corresponding to the names MULT and INV of arity 2 and 1, respectively, but they must also be smooth mappings. The degree of smoothness varies from differentiability to analyticity. As usual, three algebraic equational constraints must hold. Let A, B, and C be in M; then the constraints are as follows:

1) Associative law: (A x (B x C)) = ((A x B) x C)
2) Identity condition: I x A = A x I = A
3) One-sided Inverse: g(A) A = I is left inverse, and A g(A) = I is right inverse; both hold for Lie group. Fig. 11.2 illustrates a polyadic graph for a Lie group.

11.6 Briefing on topological manifold properties of a Lie group

Before examples are given, a review of some topological and manifold properties possessed by Lie groups will be mentioned. Manifolds M are always special topological spaces of the Hausdorff type, and are assumed to be second countable. Moreover, the Hausdorff space must everywhere be locally homeomorphic to R^n. The space appears to be locally like an Euclidean space, even though the group structure may be highly

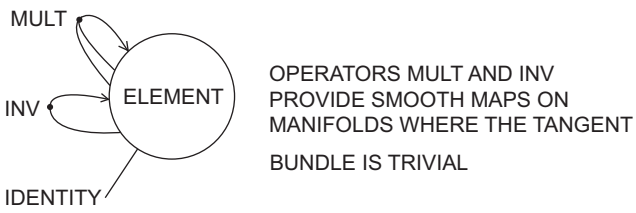


FIGURE 11.2 Lie group MSA graph.

nonlinear. A homeomorphic map is a bi-continuous, one-to-one, onto, structure-preserving map. Euclidean space is both a Hausdorff space and a second countable space.

A manifold M is a collection of open sets called charts, Ua; their union covers M. Together the charts constitute an atlas. Functions, or coordinate maps fa such that fa: M $\rightarrow$ R^n, along with fa^{-1} are homeomorphisms. The chart is called the coordinate domain, and fa is called the coordinate map. Both fa and fa^{-1} are continuous. These maps are said to be smooth whenever, fb fa^{-1} is in C^k or more often than not in C^∞. Lie groups can be viewed as an n-dimensional vector space with local coordinates. Fig. 10.5 applies one hundred percent when the Lie group is the manifold M. Just replace M by the Lie group. Here again, transition functions are homeomorphisms T, such that Tab = fb fa^{-1}; they map elements from one chart Ua into another chart Ub.

Again, some brief concepts from Section 10.3 will be reviewed for Lie group manifolds. These manifolds are a connected topological space Tp, whenever they cannot be represented as a disjoint union of two or more nonempty open sets. For a nonconnected space, there exist maximally connected subsets of the topological space Tp, called connected components. A path-connected space Tp is such that there is a continuous function f: [0, 1] $\rightarrow$ Tp, where for x, y in Tp, f(0) = x, f (1) = y. The path connected implies connected.

A topological space is compact whenever every open cover has a finite subcover. Since charts are open covers, compact manifolds need an atlas with only a finite number of charts. Compact Lie groups are important in that all irreducible representations are finite dimensional with tensor product construction (Johnson, 1976).

Example 11.8:

In this example, a general linear group will be described. First, the pure algebraic properties will be specified. Then, some of the topological and subgroup properties will be mentioned. For the sort, ELEMENTS let the carrier set be 2 by 2 invertible matrices over the real number field. Use the identity matrix for IDENTITY and the usual matrix multiplication for MULT. Let the typical matrix inversion operation correspond to INV. As seen before, the structure is a unital associative algebra, so surely it is a multiplicative group. It is symbolized by GL(2, R) and is called the general linear group of dimension four, and it is a subgroup within M(2, R) the real vector space of all 2 by 2 matrices over the real field. Subsequently, the related complex-valued Lie group GL(2, C) will be described along with its relation with the Möbius transformation group.

The general linear group GL(2, R) is a smooth manifold of four dimensions; it is a noncompact open subset of R^4. As mentioned earlier, this Lie group is represented by 2 by 2 real-valued matrices with nonzero determinants. It is disconnected and has two connected components corresponding to the sign of the determinant. Those elements with positive determinants are called the positive component and include the IDENTITY. It too is a Lie group and is denoted by GL + (2, R). Both GL(2, R) and GL + (2, R) have the same Lie algebra m(2, R). The Lie bracket in m(2, R) is the algebra multiplication [A, B], which equals AxB−BxA that can equal zero, for instance, if A = I. Thus, the Lie algebra is not a division algebra. The maximal compact subgroup for GL(2, R) is O(2, R), the orthogonal Lie group, which is described in detail in a subsequent section. For GL + (2, R), the maximal compact subgroup is the special orthogonal group SO(2) (Jacobson, 1979). These subgroups will be addressed subsequently.#

A topological space T is said to be simply connected when it is path connected, and every loop can be continuously contracted into a point in T. Rigorously, it is a homotopy between the two continuous functions f and g, describing the loop in T. Say that f: [0, 1] → T, g: [0, 1] → T, f(0) = g(0), f(1) = g(1). Then, these two paths define a loop. The properties of loops will be more thoroughly explained in Chapter 12, where homotopy is the principal concept.

Lie's third theorem states every finite-dimensional real Lie algebra is the Lie algebra of a simply connected Lie group. In essence, simple connectivity will be the criteria for establishing a 1−1 correspondence between Lie groups and Lie algebra in the real case. Finally, a Lie group is said to be simple when it is connected, not Abelian, and all closed-connected subgroups are trivial, that is, they equal the identity or the whole space.

The next figure, Fig. 11.3, illustrates two continuous paths. This shows path connectivity, but simple connectivity is problematic from an analytical or practical point of view. Here, let f(t) = t sin(π /(2t)) for t in (0, 1], and f(0) = 0. Then, f is continuous in the interval [0, 1], since it satisfies the condition at the origin that |f(t)| is less than or equal to |t|. A Lipchitz condition therefore holds for f at the origin. Also, let g(t) = 2t − t^2. Since f(0) = g (0) and g(1) = f (1), together they form a loop. However, shrinking this loop to a point, say zero would take a lot of doing because the function f is not of bounded variation. The total variation in [0, 1] equals summation, $\sum_{n=0}^{\infty}\left(1/(2n+1)\right) = \infty$. Thus, the curve corresponding to this function is not rectifiable and intuitively it is of infinite arc length.

Example 11.9:

The special Lie group SL(2, R) is a subgroup of GL(2, R). This subgroup SL(2, R) consists of all 2 by 2 matrices with determinant equal to one. It is connected, not compact, and a simple group. SL(2, R) is not simply connected; however, SL(2, C), the complex special group, will be seen to be simply connected subsequently. The dimension of SL(2, R) is three because of the constraint on the determinant. Polar form parametrization, among other parametrizations, can be employed on elements of this group. Indeed, elements of SL(2, R) can be written as the product of an orthogonal rotation matrix and a symmetric matrix with positive eigenvalues and unit determinant. The corresponding Lie algebra sl

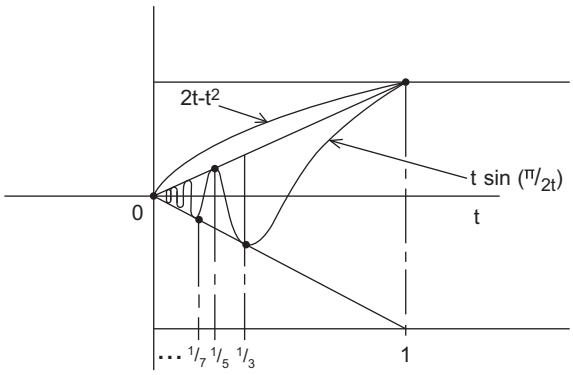


FIGURE 11.3 Path connectivity.

(2, R) consists of all 2 by 2 matrices A, and B over the reals with trace zero. The binary operation as usual is the Lie bracket commutator. So, [A, B] = AxB − BxA.

Among the many applications involving elements from SL(2, R) is the classification of conic sections utilizing a modified eccentricity parameter e, see (SL_2(R), Wikipedia). This involves the Cayley-Hamilton equation. Every matrix satisfies its characteristic equation. Thus, for A in SL(2, R), the characteristic equation for A is $x^2 - tr(A)\ x + det(A) = 0$, and since det(A) = 1, the characteristic equation becomes $x^2 - tr(A)\ x + 1 = 0$. The quadratic formula provides the solution x, for the characteristic equation:

$$x = \left(tr(A) + - \left(tr(A)^2 - 4\right)^{1/2}\right)/2. \text{ Letting } e = |tr(A)|/2 \text{ then:}$$

For e <1, A is elliptic with complex, unit, and conjugate eigenvalues, used for rotation.
For e = 1, A is parabolic with single eigenvalue +1 and −1, used as a shear operation.
For e> 1 A is hyperbolic with real reciprocal eigenvalues, used as a squeeze operation.#

Operators of the hyperbolic case mentioned in the previous example are also from the projective linear Lie group PSL(2, R). They are used as the Lorenz boost in Minkowski space. The hyperbolic functions sinh and cosh are employed in this case, providing a symmetric hyperbolic space. The latter property means that $\cosh^2 - \sinh^2 = 1$ (Bargmann, 1947).

Example 11.10:

Again consider SL(2, R). In Example 11.9, it was mentioned in not so many words that the manifold associated with this group is parametrized by the product manifold R^2 and the circle group S^1. Also mentioned is that there are other parameterizations. For instance, every point (a, b, c) in this group belongs to R^3 and a parameterization for the case where a is nonzero, which can also be given by the matrix:

$$\begin{vmatrix} a & b \\ c & (1 + bc)/a \end{vmatrix}$$

The point a is a singular point. However, when the scalar a is nonzero, the determinant of this matrix is one. Singular points are abundant in Lie groups and are problematic. In any case at these singular points, alternative parametrization is required and can often be provided.#

It seems at this point in the document all Lie groups that have been described are matrix Lie groups. An example of a nonmatrix Lie group is given next (Hall, 2013).

Example 11.11:

Again consider a dimension three Lie group, similar in dimension and construction to SL(2, R). This time, every point (a, b, c) in this group belongs to the carrier set: $R^2 \times S^1$, where S^1 is the circle group. So the third tuple is complex valued and of length one in this case. Thus c is nonzero. The real dimension in this example is four. A parameterization is given involving the addition of points in the plane R^2 and rotations

using elements of the circle group S^1. Specifically, for two elements v = (a, b, c) and w = (d, g, f) in this manifold, then

$$\text{MULT}(v, w) = \left(a + d, b + g, c\, f \exp^{iag}\right)$$
$$\text{INV}(v) = \left(-a, -b, 1/c\right), \text{note that MULT}(v, \text{INV}(v)) = \left(0, 0, c/c\right) = \text{IDENTITY}$$
$$\text{IDENTITY} = (0, 0, 1)$$

It can be shown (Hall, 2013) that there is no matrix Lie group corresponding to this abstract Lie group.#

11.7 Formal description of matrix Lie groups

Many Lie groups have elements that are highly nonlinear, and it is essential to utilize the corresponding Lie algebra to better understand the group structure. However, a simpler subset of Lie groups are matrix Lie groups. A matrix Lie group is a subset of M(n, C), where M(n, C) is the set of all complex-valued n by n matrices. Its dimension is n^2, over the complex numbers; over the reals, its dimension is $2n^2$. A matrix Ak in M(n, C) is said to converge to A in M(n, C), which means all the n^2 elements of Ak converge to corresponding elements of A. GL(n, C) is a Lie group consisting of invertible matrices. The set of all matrices A in this group form an open subset of M(n, C) and can be viewed as a manifold of dimension $2n^2$ over the real field. A subgroup G of GL(n, C) is said to be a Lie subgroup, which means that every matrix Ak in G either converges to A in G or else it is not invertible. For instance, GL(n, R) is a matrix Lie subgroup of GL(n, C) since if An is a sequence of matrices in GL(n, R) and converges to A, then the entries are real and either A is not invertible or A is in GL(n, R).

Group G is closed in GL(n, C), which means that it is closed as a subset of GL(n, C). For matrix Lie groups, connectivity is equivalent to path connectivity. The importance of the latter type of connectivity is that it is easier to validate.

Example 11.12:

Consider the special linear Lie group SL(n, R) for n greater than 1. This is the group of all n by n matrices A, with determinant one. Since the determinate is continuous, a sequence of matrices of determinate one will converge to a matrix with the same determinate value, thus showing SL(n, R) is a Lie matrix group. The matrix A can be written in polar form as the product of an orthogonal matrix and a symmetric matrix of determinant one. The dimension of this Lie group is n^2-1.#

Example 11.13:

The Lie group U (n, C) is the group of n by n complex-valued unitary matrices. Their inverse is the adjoint matrix. The dimension is n^2 over the complex field; over the reals, the dimension is $2n^2$. When n = 1, it is the circle group S^1 of all complex numbers of absolute value one. The determinant of a matrix A in U (n, C) is also a complex number of

absolute unit value. An important subgroup is SU(n, C); these are unitary matrices with determinant one. This group is of prime importance in quantum gate-type computers. The Lie algebra u(n, C) associated with U(n, C) are all skew self-adjoint matrices with the commutator as the Lie bracket.

The matrix A in U(n, C) is both connected and compact. It is compact, because in R^{2n^2}, A is closed and bounded. It is closed, because the set of points making up the real and imaginary parts of elements in A satisfy the simultaneous equations from the identity U*U = I. It is bounded because each column of A is of unit length.

Connectivity follows from path connectivity by using the unitary transform involving the matrix: V* A V = diag (e^{ip1}, . . ., e^{ipn}). Here, diag (e^{ip1}, . . ., e^{ipn}) is a diagonal matrix with each diagonal entry of unit length. Thus, A = V diag (e^{ip1}, . . ., e^{ipn}) V*. For t in [0, 1], the parametrization A(t) = V diag (e^{ipt1}, . . ., e^{iptn}) holds with t = 0 and yields A = I. So as t goes from 0 to one, A(t) makes a continuous path from I to A in U(n, C).#

Example 11.14:

Consider the carrier set consisting of elements of the special unitary Lie group of 2 by 2 complex-valued matrices SU(2, C). These matrices are of the form A =

$$\begin{vmatrix} z & -w^* \\ w & z^* \end{vmatrix}$$

where z and w are complex numbers and $|z|^2 + |w|^2 = 1$. A matrix A in SU(2, C) is unitary with determinant one. The columns of A form an orthonormal set. The inner product of the two column vectors is $<(z\ w)', (-w^*\ z^*)'> = (z^*\ w^*)\ (-w^*\ z^*)' = 0$. Moreover, the structure of the matrix A in this group is determined once the first column is defined. The second column is uniquely determined by using the orthonormality and the fact that the determinant equals one. SU(2, C) is diffeomorphic to the three sphere. This will be seen by letting z = a + bi, and w = c + di. Then, $|z|^2 + |w|^2 = a^2 + b^2 + c^2 + d^2 = 1$ gives the equation of the three sphere, S^3. When the equation for S^3 is written as $a^2 + b^2 + c^2 = 1 - d^2$, it can be seen that for varying d this becomes the equation of a point, when d = 1, along with numerous spheres for d in [0, 1). As a consequence, SU(2, C) is both compact and simply connected. Moreover, SU(2, C) is isomorphic to the unit quaternions: q = a + bi + cj + dk, where, a, b, c, and d are real, and their sum squared equals one (Gilmore, 1974).#

Previously, the carrier set utilizing SL(2, R) was illustrated with several representations. Next, illustrations will be given for the complex two-dimensional special linear group. The first example will use a continuous map D from the topological space X = SL(2, C) into a subspace. This map will also be a retraction map, wherein the operator D acts like the identity on the subspace. The subspace under these conditions is called a retract of the superspace. The mapping will also be a deformation retraction mapping D. This is a retraction procedure in which D operates on the original space to continuously contract it into the subspace. A purpose for using a deformation is to relate or extend homotopic properties from a subspace to the original space. So, if the subspace possesses certain homotopic properties, then after successfully performing the procedure, the parent space will also posses these properties. In short, deformation retract is a procedure for creating homotopy equivalence. The objective of

homotopy is to preserve or identify properties within a topological space, which are invariant under stretching or shrinking operations. The next chapter describes homotopy in detail. However, there exist spaces where deformation contracts, but does not do so in a continuous fashion, such as the zig-zig pattern. In this case, homotopy is not preserved (Hatcher, 2002).

Example 11.15:

The special linear group of 2 by 2 matrices over the complex numbers with determinant one is denoted by SL(n, C). With n = 2, the dimension is three, but over the reals, it is six. It is a simple but not compact group. Similar to SL(n, R), these matrices can also be parametrized using polar form. In this case, they can be represented as the product of a special unitary matrix, in SU(2, C), and a positive definite self-adjoint matrix with determinant one. To show SL(2, C) is simply connected, it will be shown that there exists a deformation retract D, from SL(2, C) to the subspace of special unitary matrices, SU(2, C). The subspace was seen to be simply connected in Example 11.14.#

The next example is dedicated to showing that SL(2, C) is simply connected. This will be done by illustrating that there exists a deformation retract D, from SL(2, C) to the subspace of special unitary matrices, SU(2, C).

Example 11.16:

Since SU(2, C) consists of unitary matrices, the homotopy will be accomplished by employing an arbitrary matrix in SL(2, C) and continuously transforming it into a unitary matrix. This can be done by ortho-normalizing the columns of the matrix. Thus, the Gram-Schmidt process can be employed on the column vectors. For instance, to show a retract, the left matrix below in SL(2, C) will have the second column vector transformed such that it is orthogonal to the first column. The original matrix is given next with the resulting matrix directly following:

$$\begin{matrix} |a & c| & |a & e| \\ |b & d| & |b & f|. \end{matrix}$$

In detail, the second column (c d)′ of the original matrix will be mapped into the second column (e f)′ of the second matrix by using projections involving inner products. Using the Gram-Schmidt process involving the second column gives:

$$(e \quad f)' = (c \quad d)' - <(a \quad b)', (c \quad d)'>(a \quad b)' / \left(\|a\|^2 + \|b\|^2 \right).$$

Evaluating the inner product yields:

$$(e \quad f)' = (c \quad d)' - (a \quad b)'(c\, a * + b * d) / \left(\|a\|^2 + \|b\|^2 \right).$$

The quantity $<(a\ b)', (c\ d)'>\ (a\ b)'/(\|a\|^2 + \|b\|^2)$ is called the projection of column (c d)′ onto column (a b)′ and is denoted by proj ((c d)′)

The inner product of (a b)′ with (e f)′ is also given by the product of row times column vector: (a* b*) (e f)′ and it equals zero because,

$$a^* e + b^* f = a^* c + b^* d\text{-}\left(\|a\|^2 + \|b\|^2\right)(c\, a^* + b^* d)/\left(\|a\|^2 + \|b\|^2\right) = 0.$$

Lastly, in the procedure, each resulting column in the two-by-two matrix should be normalized. The normalization is illustrated below, involving the matrix before normalization and the one after, given in order:

$$\begin{matrix} |a & e| & |a/\left(\|a\|^2+\|b\|^2\right)^{1/2} & e/\left(\|e\|^2+\|f\|^2\right)^{1/2}| \\ |b & f| & |b/\left(\|a\|^2+\|b\|^2\right)^{1/2} & f/\left(\|e\|^2+\|f\|^2\right)^{1/2}|. \end{matrix}$$

Let $x = (\|a\|^2 + \|b\|^2)^{1/2}$, $y = (\|e\|^2 + \|f\|^2)^{1/2}$, and also let:

$v = (c\, a^* + b^* d)/(\|a\|^2 + \|b\|^2)$; then, the last matrix is more efficiently written as A =

$$\begin{matrix} |a/x & e/y| \\ |b/x & f/y|. \end{matrix}$$

Since ad − bc = 1, e/y = (c − a v)/y and f/y = (d − b v)/y, so the products:

(a f)/(x y) = (a d − a b v)/(x y) and (b e)/(x y) = (b c − a b v)/(x y), subtracting gives the determinant (a d − bc)/(x y) = 1/(x y). However, to see that the determinant is one, a simple calculation A* A = I, thus providing the identity matrix.

To show the deformation retract D, it must be shown the shrinking transformation from SL(2, C) to SU(2) occurs continuously. This will be performed using columns A1 and A2 of A. Consider t in [0, 1], and use Gram-Schmidt somewhat backward; let F (A1, A2) = $(A1/\|A1\|^t\ A2/\|A1\|^t)$ for t: 0 → 1. The matrix remains in SL(2, C) and ends up with a normalized first column B1. Next, using a projection, let G(B1, A2) = (B1, A2 − t proj(A2)); this matrix is in SL(2, C) for all t in [0, 1] and for t = 1; the second column becomes orthogonal to the first column. The Gram-Schmidt orthonormal method is completed with H, the normalization of the second column, just as before. Finally, the composition of these three maps D = H G F shows the homotopy from SL(2, C) to SU(2, C). Thus, SL(2, C) is also simply connected (Gilmore, 1974).#

Example 11.17:

Again consider the carrier set consisting of elements of the special unitary Lie group SU(2, C). Following is a typical matrix A in SU(2, C), followed by the X and Z Pauli matrices:

$$\begin{matrix} |z & -w^*| & |0 & 1| & |1 & 0| \\ |w & z^*| & |1 & 0| & |0 & -1|. \end{matrix}$$

If the complex numbers z and w in the matrix A are replaced using polar coordinates, $z = r\, e^{ia}$ and $w = s\, i\, e^{ib}$, for real r, s, a, and b. Use r = cos(t) and s = sin(t), for t in [0, 2π), and note that $r^2 + s^2 = 1$. Then, the unitary matrix A can be represented using the Pauli

matrices X and Z as A = cos(t) e^{iaZ} + sin(t) X e^{ibZ}. Since Z is a diagonal matrix, so is e^{iaZ}, as well as e^{ibZ}. The matrix e^{iaZ} is given followed by matrix A:

$$\begin{vmatrix} e^{ia} & 0 \\ 0 & e^{-ia} \end{vmatrix} \quad \begin{vmatrix} \cos(t)e^{ia} & i\sin(t)e^{-ib} \\ i\sin(t)e^{ib} & \cos(t)e^{-ia} \end{vmatrix}$$

Justification for the matrix representation of e^{iaZ} is most easily seen by a power series expansion. More complicated functions of a matrix are best performed using Frobenius covariants along with the Lagrange-Sylvester expansion, described in Section 5.5.#

Example 11.18:

The Lie group O(n, R) of n by n matrices A with real elements is such that AA′ = A′A = I. Here, A′ is the transpose of A. Additionally, the columns of A form an orthonormal set. A is in O(n, R) iff <Av, Aw> = <v, w> for all v, w in R^n. The group is compact, but not connected. The determinant of a matrix in this group has values plus or minus one; these two values correspond to each of the two components. One of these includes the identity and results in the subgroup SO(n, R). As an example of matrices in the other connected component, consider the carrier set (O(2, R) − SO(2, R)). Let A be in this set; then a typical matrix is a reflection, for instance, A =

$$\begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}$$

Then, for a vector v in R^2 where v = (a b)′, then A v = (−a b), which is a reflection about the x-axis.#

Example 11.19:

The special orthogonal Lie group is SO(n, R); it is a subgroup of O(n, R) with matrices A, having determinant one. It is a subgroup that preserves the orientation of space also called a direct isometry. It is also compact and connected. But it is not simply connected (Hall, 2015). For a concrete example, consider the carrier set SO(3, R), which is a rotation group. Topologically, this space is the real projective space RP3. There are numerous charts associated with this group rendering distinct parametrizations, each leading to three-dimensional rotations. Some parametrizations involve four parameters, and some involve three, but there cannot be only a two-variable parametrization. Euler's theorem specifies any rotation in R^3 can be produced with three parameters. In this case, two angles give the axis of rotation and the third for the actual angle of rotation itself (Palais et al., 2008).

These parametrizations are valid for local use on R^3 manifolds, but there are often singularities associated with global parametrization. Among the three-parameter implementations is the Euler angle matrices. These are three matrices each producing a rotation about a coordinate axis x, y, and z, axes with parameters called the Euler angles a, b, and d, respectively. The corresponding three Euler matrices given here are specified in order of rotation axis, x, y, and z, and are provided as follows:

$$\begin{array}{lll} |1 & 0 & 0| \\ |0 & ca & -sa| \\ |0 & sa & ca| \end{array} \quad \begin{array}{lll} |cb & 0 & sb| \\ |0 & 1 & 0| \\ |-sb & 0. & cb| \end{array} \quad \begin{array}{lll} |cd & -sd & 0| \\ |sd & cd & 0| \\ |0 & 0 & 1| \end{array}$$

The notation specified here is ca, which is the cosine of a; sa represents the sine of a; and so on. All three of these rotation matrices are multiplied in some order providing a general rotation in R^3. Distinct applications of these matrices use one of the 24 different combinations due to the choice of axes of rotation, as well as the order of application of the individual matrices. Usually, this is not a problem; however, the gimbal lock is. Gimbal lock is where singularities occur in the sense that a degree of freedom is lost due to a specific matrix configuration. Technically, it occurs due to the lack of a covering map relating the three Euler angles with the rotation space. Specifically, the mapping is not a homeomorphism at certain points. The loss of a degree of freedom is illustrated utilizing the three Euler angle matrices given earlier. Prior to multiplying these matrices in the exact order from left to right, as illustrated earlier, x, y, and then z, use in the matrix for y the angle for b, as $b = \pi/2$; this gives $cb = \cos(b) = 0$ and $sb = \sin(b) = 1$. Then multiplying these matrices, x y z, yields the matrix of the product, Y. The same result would occur if the three matrices were first multiplied, and then the angle for b was substituted. The result is Y, which is given as follows:

$$\begin{array}{lll} |0 & 0 & 1| \\ |s(a+d) & c(a+d) & 0| \\ |-c(a+d) & s(a+d) & 0|. \end{array}$$

As can be seen from the above matrix Y, the three degrees of freedom for rotation are gone. When there are changes in b, there is no effect; the matrix stays the same. A primary application for the use of these types of matrices is in strap-down inertial navigation using gyroscopes and accelerometers, but no gimbals. Each of the Euler matrices is employed for either a roll, pitch, or yaw action for the craft. All these actions could range from zero to three hundred and sixty degrees.

In this application, because of the singularities with Euler matrices, quaternions are often employed instead with SCALAR equal to the reals, and VECTOR being the quaternions. Here, the quaternion for $q = a + bi + cj + dk$ is normalized, that is, $\|q\|^2 = 1$; this type of quaternion is also called a versor. Also let $s = 1/\|q\|^2$. Then, the quaternion four-parameter matrix for rotation replaces the Euler matrices. The well-known quaternion matrix for rotation is as follows:

$$\begin{array}{lll} |1-2s(c^2+d^2) & 2s(bc-da) & 2s(bd+ca)| \\ |2s(bc+da) & 1-2s(b^2+d^2) & 2s(cd-ba)| \\ |2s(bd+ca) & 2s(cd+ba) & 1-2s(b^2+c^2)| \end{array}$$

In the strap-down application just mentioned, this matrix takes the place of the mechanical gimbals. Moreover, the gyroscopes that are rigidly fastened to the craft provide the angular velocity in the form ω or actually small angle changes. This allows the quaternion matrix to change according to the derivative involving the quaternion equation: $dq/dt = -\omega q$ (Giardina, 1973).#

11.8 Mappings between Lie groups and Lie algebras

Often, there correspond several Lie groups associated with a single Lie algebra. The correspondence is very exact; from all Lie groups, a linearization can be conducted to determine the unique Lie algebra, at least in a categorial manner. On the other hand, there is an exponential map involving the Lie algebra to distinguish each Lie group. Referring to Fig. 11.4, an illustration is given of a many-to-one mapping from Lie groups to a Lie algebra. In this figure, observe that there is a linear map LIN, which goes from each Lie group to the Lie algebra. Also, there is an exponential map from the Lie algebra g to each of the Lie groups, G.

One of the principal techniques for creating the Lie algebra g corresponding to a specific Lie group G will be described later in a formal manner. It involves a topologically closed subgroup G of the general linear group GL(n, R). Closure in this development is extremely important. In this case, g = {X, such that X is in gl(n, R) where e^{tX} is in G for all t in R}. Moreover, g is the tangent vector space to G at I. Additionally, the mapping exp: g → G is locally invertible. This tangent space procedure is outlined next for general Lie matrix groups. This is similar to Example 11.4 where the tangent line is created for the circle $U(1) = S^1$. Examples are given in a formal manner, using the complex field, the real field, as well as over the quaternion skew field. Moreover, only matrix groups are considered, and all groups are assumed to be closed in the topology.

The linearization map LIN illustrated in Fig. 11.4 is obtained using the differentiation of smooth paths associated with the Lie group G. First is a smooth path p: R^1 → G, where G is the matrix Lie group, p: [− r, r] → G for some small value r. In addition, p(0) = I, the identity element in G. Next, the tangent space is formerly used. It is the set of all equivalence classes, each consisting of all smooth paths in G with the same derivative at t = 0. In a nutshell, the tangent space will be a vector space; however, it is isomorphic to what is called the Lie algebra associated with the Lie group G. A rigorous presentation of this and the following description involving tangent spaces can be found in Hall (2013).

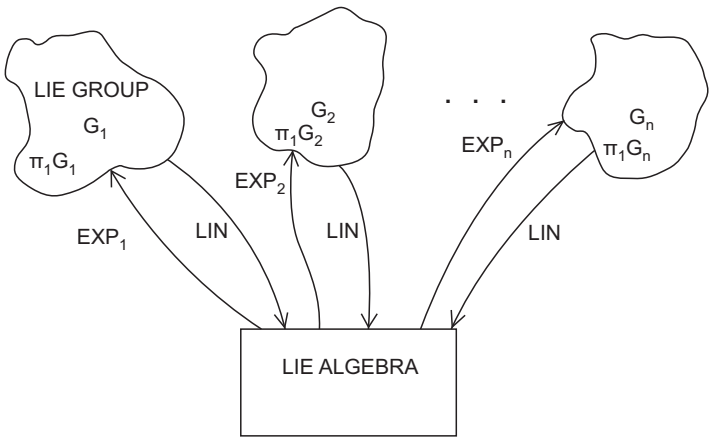


FIGURE 11.4 Mappings between Lie groups and a Lie algebra.

For a matrix Lie group G, the tangent space at the identity is denoted by T(G). Let p(t) and q(t) be smooth paths in G. Assume that the derivatives evaluated at zero are given by $dp(0)/dt = v$ and $dq(0)/dt = w$, with $p(0) = q(0) = 1$. T(G) is a real vector space. This follows by first letting v and w be in T(G). To see that v + w is in T(G), consider s(t) = p(t) q (t) then differentiating, using Leibnitz's rule: $s' = p'\,q + p\,q'$. Here, the prime indicates the first derivative. Evaluating at zero gives $s'(0) = p'(0) + q'(0) = v + w$, so the sum is in T (G). Equivalently, use $f(t) = e^{tv}e^{tw}$ in G for all real t in R, where f(0) = I. Taking the derivative df/dt gives $ve + e^{tv}we^{tw}$. Evaluating at zero gives v + w, again. To see that a real scalar c times v is in T(G), consider s(t) = p(ct), then $s'(t) = c\,p'(t)$, evaluating at t = 0 gives c v, showing this in T(G). Alternately, consider $g(t) = e^{tcv}$ in G; here g(0) = I and dg/dt evaluated at the origin gives cv; thus, T(G) is a vector space.

For any matrix Lie group G, the tangent space T(G) becomes a Lie algebra whenever the Lie bracket is the commutator: [X, Y] = X Y − Y X, for X and Y in G. The associated Lie algebra is denoted by g. To see this is a Lie algebra, the CBH formula will be employed so let $h(t) = e^{Xt^{1/2}}e^{Yt^{1/2}}e^{-Xt^{1/2}}e^{-Yt^{1/2}}$. Then using CBH gives $h(t) = e^{Xt^{1/2}} + Yt^{1/2} + 1/2[X,Y] + O\left(t^{3/2}\right)e^{-Xt^{1/2}} - Yt^{1/2} + 1/2[X,Y] + O\left(t^{3/2}\right)$, the $t^{1/2}$ terms cancel.

Applying CBH again yields $h(t) = e^{\left(t[X,Y] - t[X+Y,X+Y] + O\left(t^{3/2}\right)\right.}$, but [X + Y, X + Y] = 0. Now differentiation of $h(t) = e^{\left(t[X,Y] + O\left(t^{3/2}\right)\right)}$ gives $h'(t) = \left([X,Y] + O\left(t^{1/2}\right)\right)e^{\left(t[X,Y] + O\left(t^{3/2}\right)\right)}$. Evaluation at t = 0 gives [X, Y].

Several examples are provided illustrating the tangent space and tangent vector calculation for important Lie groups. Subsequently, Lie algebras are constructed corresponding to some of these Lie groups.

Example 11.20:

The quaternions form a Lie group SLC(1, Q). The tangent space for the quaternions at unity is the set of all vector parts. To see this, let q(t) have a smooth path with q(0) = 1. Note that $q(t)\,q^*(t) = 1$; then differentiating gives $q'\,q^* + q\,q^{*\prime} = 0$, so substituting in t = 0 gives $q'(0) + q^{*\prime}(0) = 0$. Use the scalar, q0, plus vector, v, representation for quaternions. It follows that $q'0(0) + v'(0) + q0^{*\prime}(0) + v^{*\prime}(0) = 0$, but $v'(0) = -v^{*\prime}(0)$, and therefore the real part of q = 0. So the tangent vector at 1 is a vector part of q.

The Lie algebra g for the quaternions is found using the commutator Lie bracket. Since [i, j] = 2k, [j, k] = 2i, [k, i] = 2j. This is isomorphic to the cross product algebra in R^3, [i, j] = 2 i x j, [j, k] = 2 j x k, [k, i] = 2 k x i.#

Example 11.21:

The tangent vectors pass through unity, that is, they pass through the ONE for the Lie group. In O(n, R), tangent vectors are found using matrix A(t) having a smooth path with A(0) = I. Since $A\,A^* = I$ for all t so, differentiating results in $A'\,A^* + A\,A^{*\prime} = 0$, where the prime again denotes derivative. Then, substituting t = 0 gives $A'(0) + A'^*(0) = 0$. Use the carrier set O(3, R) and as an example employ an Euler matrix A given below along with the derivative A′ as well as $A'(0) + A'^*(0)$.

Displayed in order, these matrices are as follows:

$$\begin{matrix} |\cos(t) & -\sin(t) & 0| & |-\sin(t) & -\cos(t) & 0| & |0 & 0 & 0| \\ |\sin(t) & \cos(t) & 0| & |\cos(t) & -\sin(t) & 0| & |0 & 0 & 0| \\ |0 & 0 & 1| & |0 & 0 & 0| & |0 & 0 & 0|. \end{matrix}$$

So let the tangent vector at $t = 0$ be v, where $v = A'(0)$; in any case, it follows that $v + v^*$ equals zero.#

Example 11.22:

The tangent vectors v, through unity for the Lie group U(n, C), are found using matrix B(t) having a smooth path with $B(0) = I$. Since $BB^* = I$ for all t, differentiating gives $B' B^* + B B^{*\prime} = 0$; substituting in $t = 0$ gives $B'(0) + B'^*(0) = 0$. As before, let the tangent vector at $t = 0$ be v; it is again an n by n matrix. This time, $v = B'(0)$; in any case it follows that $v + v^*$ equals zero.#

Example 11.23:

As in the previous two examples, the tangent space of SO(n, R) again consists of n by n matrices v, with $v + v^* = 0$. The tangent space here is denoted by T(SO(n, R)) and is in fact a Lie algebra using the commutator. So also consider w in T(SO(n, R)); then $w^* = -w$; the Lie bracket must also satisfy this identity. To see this, then $u^* = [v, w]^* = (v\ w)^* - (w\ v)^* = w^* v^* - v^* w^* = w\ v - v\ w = -\ [v, w] = -u$. This shows that this is a Lie algebra, and it is denoted by so(n, R). The dimension of this algebra is $n(n-1)/2$.

Use the carrier set SO(3, R), and as an example of $u^* + u = 0$, consider the Euler matrix A given below. It is followed by the derivative A′ as well as $A'(0) + A'^*(0)$, all shown in order:

$$\begin{matrix} |\cos(t) & -\sin(t) & 0| & |-\sin(t) & -\cos(t) & 0| & |0 & 0 & 0| \\ |\sin(t) & \cos(t) & 0| & |\cos(t) & -\sin(t) & 0| & |0 & 0 & 0| \\ |0 & 0 & 1| & |0 & 0 & 0| & |0 & 0 & 0|. \end{matrix}$$

For any n by n matrix M such that $M + M^* = 0$, there is the function of a matrix $A(t) = e^{tM}$, and it is in O(n, R). This can be seen formally, since M and M* commute because $M^* = -M$. It follows from a power series expansion that $(e^M)^* = eM^*$, and so $e^M\ e^{M*} = e^{(M+M^*)} = e^0 = I$. This shows that e^M is in O(n, R). Using $tM + tM^*$ illustrates the desired results. Moreover, since A(t) is continuous, this implies that the determinant is also continuous. Because A(t) is in the orthogonal group, the determinant is one or minus one. However, $A(0) = I$ shows that $\det(A(t)) = 1$ for all t.#

Example 11.24:

The tangent space of the unitary Lie group U(n, C) is the vector tangent space of all n by n matrices M such that $M + M^* = 0$. Additionally, the Lie algebra u(n, C) are all the tangent vectors in T(U(n, C)), which satisfy the Lie bracket commutator operation. The dimension of this space is n^2.#

Example 11.25:

The tangent space for the special unitary group SU(n, C) consists of all n by n matrices M such that $M + M^* = 0$ and the trace $M = 0$. This can be seen by letting $A(t) = e^{tM}$. First note that the derivative $A'(t) = M\ e^{tM}$, and substituting in $t = 0$ gives $A'(0) = M$. Consider $A(t)\ A^*(t) = e^{(tM+tM^*)} = e^0 = I$. This shows that A(t) is in U(n, C). Next, since $\det(e^{tM}) = e^{tr(tM)} = e^0 = 1$, this shows that A(t) is in SU(n, C). Going the other way since SU(n, C) is a subset of U(n, C), it follows that $M + M^* = 0$. Using the fact that e^M is in SU(n, C), implies that $\det(e^M) = I$. Using the relationship between determinants and traces gives $e^{tr(M)} = I$, and so $tr(M) = 0$. It follows that the space of all such matrices M forms the Lie algebra su(n, C) using the commutator Lie bracket. The dimension of the algebra is $n^2 - 1$.#

The process of obtaining a Lie group knowing its Lie algebra is a more difficult process than finding the Lie algebra from a given Lie group. First of all, there may be several Lie groups associated with a single Lie algebra. However, more importantly, the procedure is more complicated since it entails two steps. The first is solving a localization problem. Namely, the only thing known about the Lie group is its matrix dimension. Also, it has an identity element, say I, and the object is to find other group elements near I. This step involves an element from the Lie algebra and an isometric mapping that provides the structure of the infinitesimal matrices near the identity matrix. The following step is in finding general group elements. This step involves the basis elements from the Lie algebra along with a power series expansion of the exponential function. For simple cases, this series is summed using standard calculus techniques yielding a general element within the matrix Lie group.

The next example will illustrate how the Lie group SU(2, C) is obtained from su(2, C). It begins with u(2, C) and then finds a local representation of the group U(2, C) in terms of the Pauli matrices. Then imposing a constraint on the determinant yields a local representation for the group SU(2, C). Finally, the exponential map provides a global representation.

Example 11.26:

In this example, begin by using the Lie algebra u(2, C) of 2 by 2 skew self-adjoint matrices. To determine a Lie group having u(2, C) as its Lie algebra, take any 2 by 2 matrix M, with arbitrarily small entries. Then, $I + M = U$ is an element, that is, it is a matrix in a neighborhood of the identity of the desired group. So, set $U^* I U = I$; this equals $I + M^* + M + M^*M = I$. This is the metric-preserving property. The product term is small, and therefore setting it equal to zero results in $M^* = -M$. Thus, the matrix M has to be skew Hermitian, that is, skew self-adjoint. Accordingly, using small real numbers: p, q, a, and b, then M is of the form given below followed by $-iM$, and with slight modifications, it is again given by $-iM$. These matrices are illustrated in order; however, substitutions are made for the two real numbers p and q in the last expression. They are written as $p = c' + d'$, $q = c' - d'$, for c' and d' real:-

$$\begin{array}{llll} |\mathrm{pi} & b-\mathrm{ai}| & |\mathrm{p} & -a-\mathrm{ib}| \\ |-b-\mathrm{ai} & \mathrm{qi}.| & |-a+\mathrm{bi} & \mathrm{q}| \end{array} \quad \begin{array}{ll} |c'+d' & -a-\mathrm{ib}| \\ |-a+\mathrm{bi} & c'-d'| \end{array}$$

The last matrix will be described in terms of the identity 2 x 2 matrix I along with the three modified Pauli matrices M1, M2, and M3 specified in a previous chapter. These matrices are illustrated below in order: M1 = is1, M2 = −is2, and M3 = is3.

$$\begin{array}{llll} |0 & \mathrm{i}| & |0 & -1| \\ |\mathrm{i} & 0| & |1 & 0| \end{array} \quad \begin{array}{ll} |\mathrm{i} & 0| \\ |0 & -1| \end{array}$$

The Pauli matrices will be modified for another time involving a division by two, but first note that $-\mathrm{iM} = c'\ \mathrm{I} + \mathrm{a\ i\ M1} + \mathrm{i\ b\ M2} - d'\ \mathrm{i\ M3}$. Therefore, $\mathrm{M} = -c'\ \mathrm{iI} + \mathrm{a\ M1} + \mathrm{b\ M2} - d'\ \mathrm{M3}$.

The group SU(2, C) is a subgroup of U(2, C) but with a determinant equaling one. Take det(I + M), but only retaining linear terms gives $1 + \mathrm{pi} + \mathrm{qi}$. This results in $\mathrm{p} + \mathrm{q} = 0$, that is, $c' = 0$. As a consequence, to order one, the matrix M only involves the three Pauli matrices, not the identity matrix. As such, SU(2, C) has only three real parameters, a, b, and d′. The parameters provided the coordinates on the manifold for the Lie group. The matrix M generates any infinitesimal element in SU(2, C).

To perform the exponential mapping, structure constants and Einstein notation are useful when a general matrix in SU(2, C) is desired. Basically, M = aj Mj is a short notation for the sum $\mathrm{M} = \mathrm{a\ M1} + \mathrm{b\ M2} + \mathrm{d\ M3}$. Here, a1 = a, a2 = b, and a3 = d, all in C. So I + aj Mj generates any element in the group SU(2, C) near the identity I. A general element of the group is e^{M}, which is the summation $\sum_{n=0}^{\infty}\left(\mathrm{M}^{n}/n!\right)$. In M, substitute the three modified Pauli matrices, M1 = (is1) /2, M2 = (−is2) /2, and M3 = (is3) /2. Recall that the Pauli matrices sj have nice properties; they are such that $\mathrm{M}^2 = -\left((\mathrm{a}^2 + \mathrm{b}^2 + \mathrm{d}'^2)/2\right)\mathrm{I}$. This follows since the Poison bracket {si, sj} = 0 for i not equal to j and = 1 otherwise. Next write $\mathrm{M}^2 = -\ \mathrm{d}^2/2\ \mathrm{I}$, where $\mathrm{d}^2 = (\mathrm{a}^2 + \mathrm{b}^2 + \mathrm{d}^{2\prime})$. Now the summation for e^{M} should be broken into even and odd terms denoted by Even and Odd. Also, the indices for summation are not shown and Einstein-type notation is employed. So $\mathrm{e}^{M} = \mathrm{e}^{iajsj} = \mathrm{Even} + \mathrm{Odd}$. Utilizing the identity $\mathrm{M}^2 = -\ \mathrm{d}^2/2\ \mathrm{I}$ gives for the Even situation = I cos(d/2), and for the Odd = i(aj sj) sin(d/2) = i(a s1 + b s2 + d′ s3) sin(d/2). Using Even plus Odd expressions together provides the most general element of SU(2, C). After substituting in for the Pauli matrices sj, this yields the general matrix structure for SU(2, C); call this matrix T:

$$\begin{array}{ll} |\cos(\mathrm{d}/2) + \mathrm{id}'\sin(\mathrm{d}/2) & \mathrm{b}\sin(\mathrm{d}/2) + \mathrm{ia}\sin(\mathrm{d}/2)| \\ |\mathrm{b}\sin(\mathrm{d}/2) + \mathrm{ia}\sin(\mathrm{d}/2) & \cos(\mathrm{d}/2) - \mathrm{id}'\sin(\mathrm{d}/2)|. \end{array}$$

An important fact about this matrix is when d = 0, the identity results. However, when $\mathrm{d} = 2\pi$, minus the identity is obtained. Both the identity I and −I commute with any matrix in SU(2, C). Recall that the subgroup, {I, −I} is called the center of SU(2, C); it is the largest subgroup that commutes with all elements of SU(2, C). The group SU(2, C) is also compact since all possible values of this matrix mentioned earlier are achieved within the closed sphere in a three-dimensional manifold of radius 2π. Moreover, anywhere on the boundary of this sphere, there is only one matrix. That is, at all boundary points of the sphere, the corresponding matrices are all equal.#

Example 11.27:

To find U(2,C), most of the work is done in the previous example. The only thing to do here is to exponentiate the four basis elements from the Lie algebra u(2,C). In the previous example, $e^M = e^{iajsj}$ was found. In this example, $e^{i/2c'I+iajsj}$ need to be found. Because I commutes with all of the Pauli matrices, it follows that $e^{i/2c'I+iajsj} = e^{i/2c'I}\, e^{iajsj}$. The only thing to determine now is the first exponent $e^{i/2c'I} =$

$$\begin{matrix} |e^{ic'/2} & 0| \\ |0 & e^{ic'/2}|. \end{matrix}$$

Multiplication of this matrix with the result from Example 11.26, that is, matrix T, gives a typical matrix from U(2, C).#

Example 11.28:

Now, SO(3, R) will be found from the same Lie algebra, su(n, C) as given in Example 11.25. Again, the transformation of the metric must be invariant. In this case, since the real field is used, the transform is a similarity transform involving orthogonal matrices, O, and its transpose, O′. Hence, there exists an orthogonal matrix O such that O′ B O = B, for any matrix B in the desired group. Again, using small entries this time in 3 by 3 matrix M and substituting O = I + M yields M + M′ = 0, and so M is skew symmetric. Accordingly, using small real values a, b, and c, it follows that M =

$$\begin{matrix} |0 & c & -b| \\ |-c & 0 & a| \\ |b & -a & 0|. \end{matrix}$$

This matrix can be written in terms of a skew symmetric basis using X1, X2, and X3 all provided as follows. The first involving scalar a, the next for value b, and finally for value c:

$$\begin{matrix} |0 & 0 & 0| & |0 & 0 & -1| & |0 & 1 & 0| \\ |0 & 0 & 1| & |0 & 0 & 0| & |-1 & 0 & 0| \\ |0 & -1 & 0| & |1 & 0 & 0| & |0 & 0 & 0| \end{matrix}$$

M = aX1 + bX2 + cX3, or in terms of Einstein-type notation M = aj Xj. Note that a1 = a, a2 = b, and a3 = c in this example. Employing the Lie bracket gives [Xi, Xj] = − Eijk Xk, where the Levi-Civita symbols are again utilized. For SO(3, R), the determinant must be one. Taking det(I + M), setting it equal to one and retaining up to linear terms, just gives one. So no other constraints are needed. Again, using Einstein notation yields an actual group element using the exponential, e^{ajXj}. Before expanding this, the Cayley-Hamilton formulation of the characteristic equation is employed to obtain $M^3 + d^2 M = 0$, where $d^2 = (a^2 + b^2 + c^2)$. Similar to the previous problem, $M^3 = -d^2 M$ is used to more quickly sum the series for the exponential. Again, use even and odd terms in the series for e^{ajXj}. Continually substitute $-d^2 M$ for M^3. A simple power series approximation can be utilized in showing $e^{ajXj} = I + ajXj \sin(d) + (ajXj)^2(1-\cos(d))$. Now, substituting in for the basis

elements Xj, remembering that ajXj = a X1 + b X2 + c X3, and $d^2 = (a^2 + b^2 + c^2)$, and finally squaring ajXj and substituting gives e^{ajXj} as the following three-by-three matrix:

$$\begin{array}{lll} |1 - (b^2 + c^2)(1 - \cos(d)) & ab(1 - \cos(d)) + d^3\sin(d) & ac(1 - \cos(d)) - d^2\sin(d)| \\ |ab(1 - \cos(d)) - d^3\sin(d) & 1 - (a^2 + c^2)(1 - \cos(d)) & cb(1 - \cos(d)) + d\sin(d)| \\ |ac(1 - \cos(d)) + d^2\sin(d) & cb(1 - \cos(d)) + d\sin(d) & 1 + (b^2 + a^2)(1 - \cos(d))| \end{array}$$

This is a general element of SO(3, R). Here, notice that for d = 0, the identity matrix I is obtained. This corresponds to the origin of the three-dimensional manifold representation. When $d = \pi$, $e^{ajXj} =$

$$\begin{array}{lll} |1 - (b^2 + c^2)(2) & ab(2) & ac(2)| \\ |ab(2) & 1 - (a^2 + c^2)(2)) & cb(2)| \\ |ac(2) & cb(2) & 1 + (b^2 + a^2)(2)|. \end{array}$$

This matrix also holds for antipodal points in the manifold. In this case, $d = \pi$, and the coordinate variables a, b, and c are now all negative. Under these conditions, the exact same matrix is obtained when a, b, and c are positive. Any straight line segment through the origin intersects a sphere of radius π at two opposite points. These are antipodal points, and the matrices e^{ajXj} at these two points are identical. Additionally, this Lie group is compact because all possible values of the general matrix are obtained in this closed sphere of radius π.

Example 11.29:

It will be seen that the special group of one-dimensional quaternions, SLC(1, Q), also has the same Lie algebra as the two different groups given in the previous examples. That is, SU(2, C) and SO(3, R), and the group SLC(1, Q) all have the same Lie algebra, su(2, C). For SLC(1, Q), the elements here are one by one matrices. The identity for the quaternions is I equal to 1, so the vector part v of q is zero. A matrix near the identity is I + q where q is a small in absolute value quaternion say q = a0I + a1i + a2j + a3k. In Einstein notation, v = aksk. The Lie algebra glc(1, Q) has a basis of four elements. This is the general linear algebra. The special Lie algebra consists of normalized quaternions. A change of basis for this Lie algebra yields X1 = −1/2i, X2 = −1/2j, and X3 = −1/2k. So [X1, X2] = 1/4 (ij−ji) = 1/2k = X3. Consequently, [Xi, Xj] = − EijkXk. For SLC(1, Q), a norm of one is required, so (1 + q) (1 + q)* = 1. Multiplying gives $1 + q + q^* + |q|^2 = 1$, which implies since v = −v* and retaining only linear terms results in a0 = 0, only the vector part of small q is left.

Now to find a general element of SLC(1, Q), take $e^{(akXk)}$; this is similar to what was done in the previous examples. Note that $(ak\ Xk)^2 = (a1X1 + a2X2 + a3X3)^2 = (a1X1)^2 + (a2X2)^2 + (a3X3)^2 = -1/4\ ((a1)^2 + (a2)^2 + (a3)^2) = -1/4\ d^2$, where $d^2 = (a1)^2 + (a2)^2 + (a3)^2$) as in previous examples. Using even and odd terms of the power series as before gives $e^{akXk} = \cos(d/2) + 2akXk/d\ \sin(d/2) = \cos(d/2) - (a1i/d + a2j/d + a3k)\sin(d/2)$.

If GLC(1, Q) is desired, then all that needs to be found is $e^{aOI+ajXj}$; this is performed in the next example.

Example 11.30:

Imitating the results from Example 11.27, the exact same procedure is utilized. The Lie group GLC(1, Q) can be obtained by employing the four basis elements in glc(1, Q) as mentioned earlier. The result is found by multiplying the exponentials e^{aOI} e^{aXj}; this results in the expression e^{aOI} e^{aXj} = a0(cos(d/2) − (a1i/d + a2j/d + a3k)sin(d/2)).

11.9 Complexification of Lie algebras

It was seen that a real vector space can become a complex vector space by complexification. Indeed, the carrier set for the sort SCALAR was swapped from the real numbers to the complex numbers. Correspondingly, the complex field was utilized in place of the real number field. Concurrently, carrier sets for the sort VECTOR were adjusted to include imaginary parts. Operators from the signature sets with names S-MULT and V-ADD were used in mapping the new elements in accordance with their definitions. In short, scalar multiplication and vector addition are closed using complex manipulations. Since the tangent space is a vector space, the vector space complexification is applicable. Similar to a Hilbert space situation, the tensor product Hc = Hr ⊗ C is applicable.

For a Lie algebra, the Lie bracket must also be complexified. In this case, for A, B, D, and E in the Lie algebra g, then the complexification of the Lie bracket is given by a form similar to the polarization identity: [A + i B, D + i E] = [A, D] − [B, E] + i[A, E] + i[B, D] (Humphreys, 1997). Finally, realize that the matrix A + Bi is shorthand for A⊗1 + B⊗i, where ⊗ is the tensor product. As a result, complexification is similar to just complexification of a vector space. An excellent example follows using a Lie algebra, which is falsely assumed to be a complex algebra.

Example 11.31:

The Lie algebra su(n, C) is the algebra with Lie bracket being the commutator consisting of all matrices M, such that M* = −M. These matrices are skew Hermitian, or skew symmetric and with a trace of M equaling zero. Note that this Lie algebra is not complex; it is real, because as a vector space it is real. This follows using M, then (Mi)* = − M*i = Mi. Refer to the vector space definition. Here, the SCALAR sort determines whether the vector space is real or complex. In any case, the most general matrix in su(2, C) is M =

$$\begin{vmatrix} ix & -c^* \\ c & -ix \end{vmatrix}$$

where x is real and c is a complex number. A basis for this Lie algebra is given next, and they are related to the Pauli matrices, sj, j = 1, 2, and 3. Given below in order are M1 = is1, M2 = −is2, and M3 = is3. These are also denoted X, Y, and Z:

$$\begin{vmatrix} 0 & i \\ i & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix} \quad \begin{vmatrix} i & 0 \\ 0 & -i \end{vmatrix}$$

The commutators [M1, M2] = 2 M3, [M2, M3] = 2 M1, and finally, [M3, M1] = 2 M2. [Mi, Mj] = 2 Eijk Mk. Here, the Levi-Civita symbol Eijk is once again employed. The Lie algebra sl(2,C) = su(2,C) + su(2,C) i. This expression will provide the complexification of su(2,C). The most general element in the special linear algebra, sl(2,C), is N =

$$\begin{vmatrix} x+iy & u+iv \\ r+is & -x-iy \end{vmatrix}$$

This matrix has complex values, and the trace equals zero. A basis for this Lie algebra is given in the order, N1, N2, and N3 as follows:

$$\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \quad \begin{vmatrix} 0 & 1 \\ 0 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & 0 \\ 1 & 0 \end{vmatrix}$$

Going back to su(2, C) and now utilizing the above basis shows that the most general matrix in su(2, C) is M = ixN1 − c*N2 + cN3, since M is given by:

$$\begin{vmatrix} ix & -c^* \\ c & -ix \end{vmatrix}$$

This is not surprising since su(2,C) is a sub-Lie algebra of sl(n, C). The objective is to see the expansion involving the complexification of su(2, C). Now consider the most general matrix in sl(2, C), which is N = −ixM3 + yM3 − u/2 (M2 + i M1) − iv/2 (M2 + i M1) + r/2 (M2 − iM1) + is/2 (M2 − iM1). This shows that the two Lie algebras are essentially the same. Moreover, note that only six complexified basis elements of su(2, C) are needed in describing N. These six matrices are M1, M2, M3, iM1, iM2, and iM3. Again, this is expected since the dimension of sl(2, C) is six over the real field.#

References

Bargmann, V., 1947. Irreducible unitary representations of the Lorentz group. Ann. Math. 48n (3).
Benkart, G., 1977. On inner ideals and ad-nilpotent elements of Lie algebras. Trans. Am. Math. Soc.
Berry, D., 2006. Efficient quantum algorithms for simulating sparse Hamiltonians, arXiv:quant-ph/0508139.
Draper, C., Meulewaeter, J., 2022. Inner ideals of real simple Lie algebras. Bull. Malaysian Math. Sci. Soc. 45.
Giardina, C., 1973. Strapdown navigation with Cayley transforms and quaternions, Singer Kearfott, report.
Gilmore, R., 1974. Lie Groups, Lie Algebras and Some of Their Applications. J Wiley & Sons.
Hall, B., 2013. Quantum theory for mathematicians978-1461471158 GTM, Vol. 267. Springer.
Hall, B., 2015. 978-3319134666 second ed. Lie Groups, Lie Algebras, and Representations, Vol. 222. Springer.
Hatcher, A., 2002. 0-521-79540-0 Algebraic Topology. Cambridge U. Press.
Humphreys, J., 1997. 978-3540900535 second ed. Introduction to Lie Algebras and Representation Theory, Vol. 9. Springer.
Jacobson, N., 1979. 9978-0486-63832-4 Lie Algebras. Dover.
Johnson, R., 1976. Representations of a compact groups on topological vector spaces: some remarks. Proc. Am. Math. Soc. 61.
Palais, B., et al., 2008. A disorienting look at Euler's theorem on the axis of rotation. Am. Math. Monthly 116 (10), 200.

CHAPTER

12

Fundamental and universal covering groups

12.1 Homotopy a graphical view

Recall that a topological space T is said to be simply connected when it is path connected. That is, there exists a continuous function, f: [0, 1]→T, such that for every two points, x and y in T, f(0) = x and f(1) = y. For another continuous function g: [0, 1]→T, where f(0) = g(0) and f(1) = g(1). Then, these two paths define a loop when they can be continuously contracted into a point xo, in T. In this case, f(0) = f(1) = g(0) = g(1). It is termed a homotopy, H between the two continuous functions f and g, providing the loops in T (Whitehead, 1978). The homotopy must be such that:

H: [0, 1] x [0, 1]→T, where
H (t, 0) = xo, for all t in [0, 1]; here, xo is the initial as well as the final point in T;
H (t, 1) = xo for all t in [0, 1]; t is a parameter positioning a loop within T;
H (0, s) = f (s) for s in [0, 1], t = 0; this positions f, and by varying s, it creates the loop;
H (1, s) = g (s) for s in [0, 1], t = 1; this positions g, and varying s, it creates the loop.

Refer to Fig. 12.1; the unit homotopy square [0, 1] x [0, 1] performs the homotopy. Vertical lines within [0, 1] x [0, 1] map into and create different continuous loops in T. Values of t on the horizontal bottom of the square locate continuous functions that correspond to loops. These values only act as pointers to a loop. At the extreme vertical edges, on the left of the homotopy square t = 0, there is the mapping creating loop A. This is formed by the continuous function f. It will be written f(s) = A, so as s changes from zero upward toward the top of the square toward s = 1, the loop forms. On the right-hand side of this square, that is, at t = 1, the function g is indicated; it provides the loop B. Thus, g(s) = B. Other values of t determine other loops that are not necessarily in between loops A and B in T. Other vertical lines lie in between the lines for f and g inside the unit square. In the diagram, the continuous function h(s) = C occurs at t = 1/3. Note that curve C overlaps curve B. Using H(t, s), homotopy is demonstrated by continuously varying t and then showing that the loops formed by H(t, s) also vary continuously from H(0, s) to H(1, s).

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00025-4

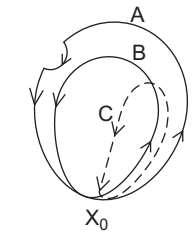
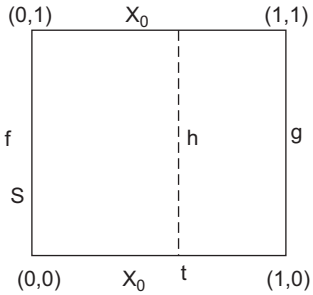
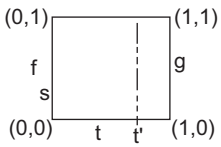


FIGURE 12.1 Homotopy square.

f : 0 ⟶ 1 A
g : 0 ⟶ 1 B
h : 0 ⟶ 1 C





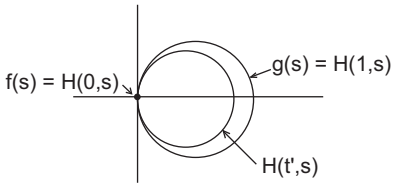


FIGURE 12.2 Example of homotopy with loops touching only at the origin.

Example 12.1:

Let $T = R^2$, and let xo be the point (1, 0). Assume that f(s) = (cos(2πs), sin(2πs)), for s in the interval [0, 1]. In this case, as s goes from zero to one, f defines the loop being the unit circle centered at the origin. It starts at xo = (1, 0) and traverses the circle, coming back to xo.#

The next example provides a homotopy where the loop-determining parameter t does yield loops in between those given by f and g. See Fig. 12.2.

Example 12.2:

Again let $T = R^2$. Also use xo = (0, 0). Let f be a constant fixed-point identity map, that is, f (s) = (0, 0) for all s in [0, 1]. Also let g (s) = (1−cos (2πs), sin (2πs)). The operation H shows that f and g are homotopic. Moreover, by continuously varying t from zero to one, the function H (t, s) provides continuous distinct loops between those given by f and g. To see the homotopy, use H (t, s) = (t (1 −cos (2πs)), t sin (2πs)) and vary t from zero to one continuously. As usual, the value t points to distinct loops.#

12.2 Initial point equivalence for loops

The fundamental group often denoted by π_1 is found only using topological properties involving connectivity. It is, however, an actual algebraic group describing topological shapes. In particular, it specifies holes within simple topological structures such as path-connected spaces. This is illustrated subsequently. Higher homotopy groups are used in finding holes within more general topological structures such as high-order spheres. Actually, the fundamental group consists of equivalence classes satisfying RST relations involving loops that can be continuously converted into each other. See Example 12.3. This is designated as a homotopy; it was previously defined in reference to paths and connectivity. That homotopy forms RST equivalence classes is shown next.

To prove reflexivity, that is, f is homotopic to itself in the homotopy square, let H(0, s) = H(1, s) = f(s), but also let H(t, s) = f(s) for all t in [0, 1]. This is the constant map, it begins with f(s), and as t: 0→1, it stays at f(s) and winds up at f(s). Next, to show symmetry, if f is homotopic to g, then there is a homotopy map H(t, s). To show that g is related to f, use H(1 − t, s). Accordingly, as t: 0→1, H(1, s) = g goes to H(0, s), which is f. Finally, to show transitivity, say that F(t, s) provides a homotopy from f(s) to g(s). Also, say that G(t, s) gives a homotopy between g(s) and h(s). Then, H(t, s) will give a homotopy between f(s) and h(s) when H(t, s) = F(2t, s) for t in [0, 1/2], and H(t, s) = G(2t − 1, s) for t in [1/2, 1]. Note that at t = 1/2, F(1, s) = G(0, s), which represents the same point xo. The next example will illustrate equivalence classes of homotopy.

Example 12.3:

Consider the x, y plane in R^2, but with a pinhole at (0, 0). This can be modeled as if a post existed at the origin, and strings are used to create loops. See Fig. 12.3. Use any point xo different from (0, 0) as a starting and ending point. Loops that do not enclose the pin are homotopic to just xo; this acts like the identity. These loops are all homotopic to each other and form an equivalence class denoted by [I]. Later this coset will employ a coset leader of zero. Loops belonging to [I] can be traced in the positive direction, that is, counterclockwise, or in the reverse direction, clockwise. For this coset, it does not matter; these loops are all equivalent. All such loops are homotopic to xo. This is because all these loops can be continuously shrunk to xo. Loops not equivalent to the identity must again start and end at xo but must now also enclose the pin. These loops are not homotopic to the

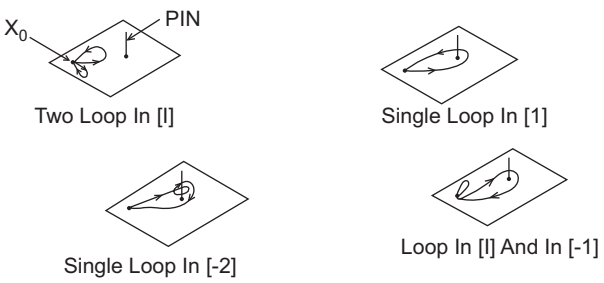


FIGURE 12.3 Equivalence classes for loops.

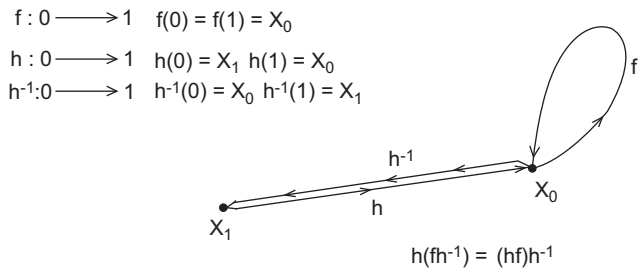


FIGURE 12.4 Initial point equivalence.

loops in [I]. A continuous tightening of a string starting and ending at xo and enclosing the pin can never morph totally into the point xo due to the post at the origin being in its interior.

There are distinct equivalence classes for the number of times the loop encloses the pin. The pin is pointed upward from the level surface. For the counterclockwise enclosures, these equivalence classes are denoted by [1], [2], [3],... indicative of the number of times the loop fully encloses or entirely wraps around the pin. Thus, there exist an infinite number of equivalence classes. Moreover, direction matters here. So, if a loop g goes clockwise and encloses the pin only once, it cannot belong to [1] because no homotopy can be found between f in [1] and g. In this case, g belongs to [− 1]. Similarly, if a loop encloses the pin n times going clockwise, then it belongs to the equivalence class [− n]. This is illustrated in Fig. 12.3. The result is that there exists an equivalence class for each integer with integer 0 corresponding to I in [I].#

The set of all equivalence classes of loops form a group under the concatenation of functions. For a path-connected space, loops need not occur at the same point xo; here, loops with different starting points are homotopic. In this case, homotopy is shown by traversing the path connecting distinct initial points in two ways. Fig. 12.4 illustrates the process. In this diagram, the path from initial point x1 to xo is traversed by h and the loop from xo back to xo by f. The trace then returns to the point x1 using the function h^{-1}. Homotopy follows by letting x1 approach xo continuously. The total path is given by h x (f x h^{-1}) or (h x f) x h^{-1}. Using the associativity property of groups, these paths are the same up to homotopy. The associative law is proved later.

For a Lie group, the fundamental group is always abelian (Humphreys, 1997). However, this is not the case in general; a counterexample is provided in Example 12.10.

12.3 MSA description of the fundamental group

In the MSA description, the carrier set for the single sort ELEMENT is the set of all equivalence classes of loops. Equivalence classes should be written [f], designating all loops homotopic with f, but this will not be the case in this section. It should be understood that equivalence classes are employed without the usual notation, that is, just use f instead of [f]. Also in the diagrams, loops as well as their defining functions are denoted by the symbols expressing the continuous functions f and g, creating these loops. The

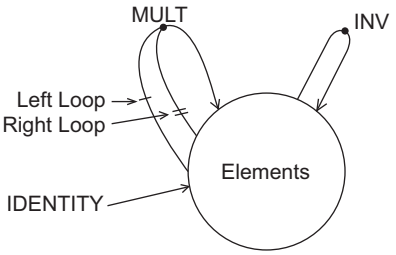


FIGURE 12.5 Polyadic graph for the fundamental group.

symbol ~ is used to symbolize equivalence. The MULT here is concatenation not composition of functions. Thus, the order of operation in the binary case is that the left loop or operand is employed first; then, the right argument is employed in MULT. The polyadic diagram in Fig. 12.5 is a reminder of the employment of the order of binary operations. As usual, the tail of the polyadic arrow has a single dash for the first operand. The second tale of MULT in the polyadic arrow is marked with two slashes for the second operand. All the group operational names within the signature sets are as follows:

MULT(f, g): [0, 1] x [0, 1] → T, where T is a topological space.
MULT(f, g)(s) = f(2 s) for s in [0, 1/2] and = g(2 s -1) for s in [1/2, 1].
INV: f^{-1} (s) = f(1 − s) for s in [0, 1].
IDENTITY: e(s) = xo for all s in [0, 1].

The usual equational identities must hold for the group structure. Here, for convenience, replace or use:

ELEMENT by A, B, C.
MULT(A,B) by A x B.
IDENTITY by I.
INV(A) by A^{-1}.

The equational identities are written below as strict identities, but they actually hold under homotopy involving equivalence classes; they do not hold point-wise unless by coincidence:

1) Associative law: (A x (B x C)) = ((A x B) x C).
2) Identity condition: I x A = A x I = A.
3) Inverse: A^{-1} x A = I is left inverse, and A x A^{-1} = I is right inverse; both must hold for a group structure.

The polyadic graph in Fig. 12.5 illustrates the operator names used in homotopy.

Fig. 12.6 illustrates that if f ~ f′ and g ~ g′, then MULT(f, g)(s) ~ MULT(f′, g′)(s). In this figure, there is a path-connected topological space T, with equivalent loops f and f′ with homotopy H, along with equivalent loops g and g′ with homotopy K. Also in this diagram is the homotopy square with K on top of the interior, above s = 1/2, and H on the bottom half of the interior of the homotopy square. On the left side is g on top and f on the bottom. On the right side is g′ on top and f′ on the bottom. Here, MULT(f, g)(s) = f(2 s) for s in [0, 1/2], and = g(2 s −1) for s in [1/2, 1]. This essentially shows that for equivalence

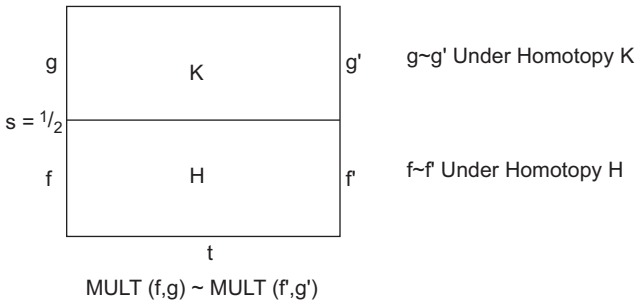


FIGURE 12.6 Equivalent classes are well defined.

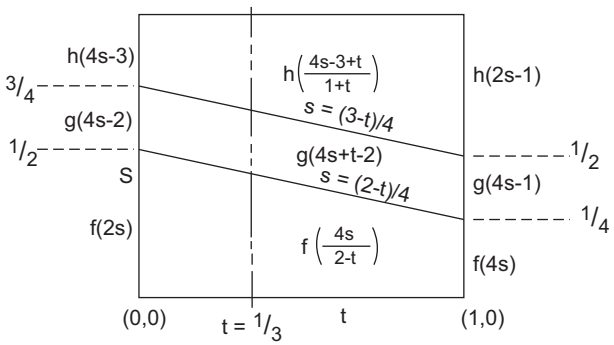


FIGURE 12.7 Proof of the associated law.

classes the concatenation type of multiplication is well defined. Note that at $s = 1/2$, $f(1)$ equals $g(0)$; this indicates that they are equal and, in fact, equal to xo.

The associative law is proven in Fig. 12.7. In this diagram, there is a topological space T, with three loops. These are actually equivalence classes. The continuous functions f, g, and h denote the loops. In this case, $f(0) = f(1) = g(0) = g(1) = h(0) = h(1)$. View the two distinct vertical sides of the homotopy square. On the left side of the square, $t = 0$, the product f x (g x h) is illustrated. On this side, let $H = f(2\,s)$ for s in [0, 1/2], $H = g\,(4\,s - 2)$ for s in [1/2, 3/4], and $H = h(4\,s - 3)$ for s in [3/4, 1]. On the right-hand side, the product (f x g) x h is given. Here, $K = f\,(4\,s)$ for s in [0, 1/4], $K = g(4\,s - 1)$ for s in [1/4, 1/2], and $K = h\,(2\,s - 1)$ for s in [1/2, 1]. Again, the associative law holds for the equivalence classes and not for the individual functions f, g, and h. It holds under homotopy, not point-wise. If any point t in [0, 1] is chosen, then this point determines the specific functions forming the homotopy. Notice that there are straight line segments separating f, g, and h, where they are defined in terms of s and t. In addition, the arguments for these functions are for the most part nonlinear.

At $t = 0$, on the left side of the homotopy square, the evaluation of the functions illustrated agrees with H given earlier. The same is true for $t = 1$ on the right-hand side; evaluation gives K. Additionally, for any point t in (0, 1), the homotopy exists because the transitions are continuous. For instance, as seen using the homotopy square, at $t = 1/3$, there exists a dashed vertical line. On this dashed line, three distinct ranges for the domains exist as s increases from zero to one vertically. There is one domain for each of f, g, and h using

the independent variable s. For f, the domain is [0, 5/12]. For g, the domain is [5/12, 8/12], and for h, the domain is [8/12, 1]. The evaluation of these functions f, g, and h, which of course are representatives of equivalence classes of concatenated loops, will show the homotopy. On the lower inclined line segment, and at t = 1/3, here s = 5/12, and evaluating f gives f((20/12)/(2−1/3)) = f (1). For this same point (t, s), evaluating g results in g(20/12 + 1/3−2) = g(0). Note that f(1) = g(0) are equal in terms of homotopy; these represent equality in terms of the final and initial points. That is, they denote the same point xo in T. In a similar way, on the upper incline line segment for t = 1/3, s = 8/12. At this point, the evaluation of both functions f and g yields g(1) = h(0).

The proof that the identity condition holds for the equivalence class is provided in Fig. 12.8. Only one part of this identity, f x I = f, is shown in the diagram. The other part (I x f = f) is proved analogously. Illustrated in this figure is the homotopy square. On the left is f x I, and on the right is just f(s). The homotopy is obtained first by considering the left-hand side, that is, t = 0. Here, H = f(2 s) for s in [0, 1/2], and H = f(1); that is, it is a constant for s in [1/2, 1]. On the right-hand side is f(s) alone. For any t between 0 and 1, the lower region of f has an upper boundary equal to the incline line segment, s = (1 + t)/2. In this region, the argument of f is 2 s/(t +1); thus, for any t and corresponding s on the incline, the upper boundary values of f agree; it is f(1). This is the same value as above the incline.

Fig. 12.9 involves the inverse function for f. The purpose of this diagram is to illustrate the fact that MULT(f, INV(f)) = IDENTITY. This is the right-sided identity. Left-sided identity relation can be shown in a similar manner. The identity in the topological diagram is the point xo. Any loop in this diagram may go all the way from xo to xo and then unwind and go back again indicative of the inverse function. However, this only happens at t = 0. In all other cases, the path will start from xo, but it will stall. It essentially will stop and then turn around and go back, reaching xo again. This indeed is the case illustrated in the Fig. 12.9 below. Note that counterclockwise motion is attributed to f and clockwise motion is attributed to INV(f). At the upper left corner, the small diagram within this figure illustrates a full revolution xo to xo. This is followed by an inverse full revolution. This correlates only with the left boundary of the homotopy square, t = 0, and was previously explained. Partial revolutions are illustrated underneath this figure in the upper left-hand corner. Note that in this application INV(f)(s) = f(2−2 s).

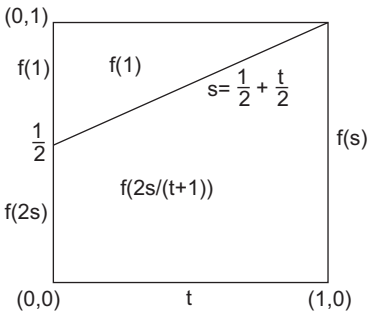


FIGURE 12.8 Identity condition.

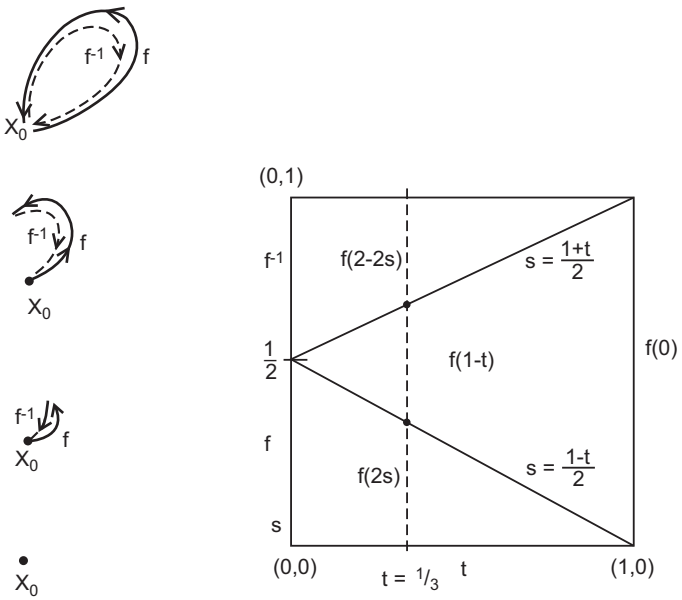


FIGURE 12.9 Inverse function equational identity.

The homotopy square H is also given in the figure. On the left-hand side of H are f and f^{-1}. Specifically, for s in [0, 1/2], f(2 s) is applied. In the interval [1/2, 1], f(2−2 s) is used. On the right side of the homotopy square is f(0), indicative of the identify function. For t in [0, 1], there exist inclined line segments $s = 1/2 + t/2$ and $s = 1/2 - t/2$ for s in [0, 1]. These line segments provide boundaries for the arguments of the homotopy function f. For fixed t, values of s in [0, 1] result in a loop or a partial loop for f. In particular, at the extremes of the square, to the left as s: $0 \to 1/2$, f makes an entire loop from xo to xo counterclockwise. For s: $1/2 \to 1$, the loop backtracks on itself in a clockwise fashion. This is illustrated in the uppermost diagram to the left of the homotopy square. On the right-hand-side boundary of the homotopy square is the identity, where for s in [0, 1], no movement occurs, and the loop stays at the point xo, for all s in [0, 1]. However, for t in (0, 1), in all cases movement does occur. However, it does not make a full loop counterclockwise, and movement stalls and proceeds clockwise back to xo.

For instance, at $t = 1/3$, there is a dashed vertical line in the diagram. This line intersects the v-type inclines at two points. On the lower incline segment, the intersection provides $s = 1/3$. Evaluation of the function f in the lower region at $s = 1/3$ gives f(2/3). Evaluation again, this time from the central region at $s = 1/3$, yields $f(1-1/3) = f(2/3)$, which is the same value again. This is a must for continuity. In this case, a loop is not attained. There is no movement, as s changes upward, until the upper inclined line segment is hit; that is, as s increases, s: $1/3 \to 2/3$. In the uppermost region, movement starts again but clockwise. It starts at the point (1/3, 2/3) providing the value $f(2-4/3) = f(2/3)$, agreeing with the value of f interior to the value of f in the v-shaped region. On the upper region, counterclockwise motion prevails, and finally, f winds up at f(1), the identity. As t: $0 \to 1$, continuous changes in loop creation occur, from a full loop and its undoing continually to just a point. The sequence is illustrated to the extreme left of the homotopy square in Fig. 12.9.

12.4 Illustrating the fundamental group

The fundamental group will be found for several situations, and in particular for Lie groups. These groups are isomorphic to well-known infinite groups such as Z, under addition, or the positive reals under multiplication. Another fundamental group of infinite cardinality is described next in the following example. Here, equivalence classes are again denoted using the usual bracket notation, [].

Example 12.4:

Refer to the x, y plane in R^2, but with a pinhole at (0,0), as given in Example 12.3. The identity corresponds to the equivalence class [I] of all loops not enclosing the origin and is homotopic to point xo. Loops enclosing the origin n times counterclockwise are in equivalence class [n]. For those loops containing the origin and enclosing it m times clockwise, form the equivalence class [− m]. The corresponding fundamental group is isomorphic to Z, all the integers under usual addition. For instance, if f is in [2] and g is in [3], then MULT(f, g) is in [5]. Additionally, if h is in [I], then MULT(f, h) is in [2], and if k is in [− 7], then MULT(f, k) is in [− 5]. Finally, INV(k) is in [7].#

The next few examples are presented in an intuitive manner. For rigorous presentations see (Strom, 2011; Humphreys, 1972; Hatcher, 2002).

Example 12.5:

For the Lie group SO(3,R), as presented in Example 11.28, it was seen that the closed sphere of radius π is the full domain for a typical matrix e^{iaXj} of SO(3,R). That is, the coordinates on the manifold corresponding to any element of this group lie in or on this sphere. Additionally, identical matrix values exist at the two points, namely at any point along with the corresponding antipodal point both on this sphere. Because of this, the fundamental group associated with SO(3,R) has more elements than just the equivalence class containing the identity I. Taking any two loops f and g, starting and ending at interior point xo and staying in the open region of this sphere, then f ∼ g ∼ I. However, for any loop h, starting at xo if it touches the boundary at any point p, it goes back to xo, leaving from the antipodal boundary point, which is directly opposite from point p. The reason is that the values of the matrix at this point and at the antipodal point are identical. They represent the same element of the group. Because of this, a new equivalence class forms; assume it contains -I. In short, the fundamental group is isomorphic to Z2. It has cardinality two. Note that two distinct paths beginning at xo and touching the sphere result in two antipodal points. The corresponding paths then return to xo, meaning that they are in the equivalence class [I].#

Example 12.6:

For the Lie group SU(2,C), as described in Example 11.26, it was seen that the closed sphere of radius 2π is the full domain for a typical matrix e^{iajsj} of SU(2,C). Additionally, at every point on the boundary of this sphere, there is only one corresponding matrix. The

identity homotopy holds for all loops in SU(2,C). That is, there is only one equivalence class; it is the identity. Begin with two loops f and g, both starting and ending at xo. Also assume that f stays in the interior of the sphere and returns to xo. However, let g hit the boundary and return to xo, leaving some arbitrary point p on the boundary. In this case, all points on the boundary act like a single point. This is because substituting in boundary coordinates yields a group element, that is, a matrix that is unique. The corresponding loops can always be brought back to point p and detach. The conclusion is that $g' \sim f'$. Consequently, the fundamental group only contains one equivalence class [I]. The difference between this example and the last one is that here the points on this sphere are identical, so movement along the boundary is permitted. In the previous example, once movement on the surface of the sphere occurs, this always corresponds to the identical-type movement of the antipodal point.#

Example 12.7:

In R^2, the figure eight is a wedge sum of two circles. The fundamental group is the non-abelian free group consisting of two generators. Generator a winds around clockwise always on the same half of the figure eight, the left half. Also, generator b winds around counterclockwise on the other half of the figure eight. These are positive rotations on these circular-type surfaces. When a winds counterclockwise on the left half, it forms inverse-type words; one complete revolution forms $INV(a) = a^{-1}$; similarly, for the right circle with b. See Fig. 12.10. The free group of two generators consists of words or elements of the form: $(v = a^{n1}\ b^{m1}\ \ldots a^{nk}\ b^{mp})$, ni, and mj are the integers. These form strings of words as rotations occur around the figure eight. The $IDENTITY = a^0 = b^0$, denoted by I. For instance, the string $f = a^2b^3$ corresponds to two rotations to the left clockwise, followed by three rotations counterclockwise to the right. Now assume that the multiplicative process and inversion operations are extended to strings. Then it follows that $MULT(f, INV(f)) = a^2b^3 \times (b^{-3}a^{-2}) = I$.#

A more in-depth description of an algebra consisting of string manipulations is given in Section 14.4, entitled MSA For Partial Isometries.

12.5 Homotopic equivalence for topological spaces

Two topological spaces X and Y are said to be homotopically equivalent whenever $F: X \to Y$, $G: Y \to X$, and G(F) = the identity on X, and F(G) is the identity on Y. When this holds, the notation given for F and G is homotopically equivalent, $F \sim G$. Under this condition, the equalities for the identities are given by G(F) = Ix and F(G) = Iy, respectively. This

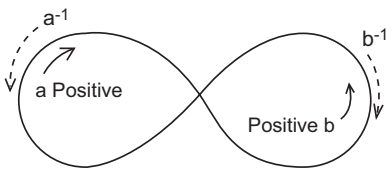


FIGURE 12.10 Figure eight.

equivalence obeys the RST relations. Note that the operation in this case is function composition and not concatenation. The functions equaling the one-sided identity when composed are called homotopic inverses. When two topological spaces are homeomorphically equivalent, they are homotopically equivalent. The converse is not true.

Example 12.8:

Consider the topological spaces $X = R^n$, and say that the other space Y is just some point in R^n, say that $Y = \{0\}$. Then F: $X \rightarrow Y$, so all points in R^n map to 0, and say that G: $Y \rightarrow X$ is such that $G(Y) = \{0\}$. So $G(F) = Ix$ and $F(G) = Iy$. The two spaces X and Y are homotopically equivalent and not homeomorphic.#

12.6 The universal covering group

For a connected topological space X, there exist a simply connected space Y and a map f, such that f: $Y \rightarrow X$. This mapping is called a covering mapping. When X is simply connected, it has a trivial fundamental group and it is its own universal cover (Hatcher, 2002). A universal covering group G for a connected Lie group H is the unique simply connected Lie group among all groups with the same Lie algebra h, as H. These groups form a family of groups having h as their Lie algebra. A Lie algebra has a special associated simply connected Lie group, which becomes the universal covering group. As with every covering group, there exists a kernel K for a covering homomorphism L. The homomorphism is a continuous, onto group homomorphism L: $G \rightarrow H$. Also, K is a discrete normal subgroup of G, and $H = G/K$. Points mapped to p in H are called fibers over p; they are left cosets. Additionally, disjoint open sets enclosing p, an interior point in H, are called sheets.

Example 12.9:

Consider the circle group T. It is a one-dimensional manifold and a single-parameter subgroup of the group of nonzero complex numbers under multiplication. Specifically, $T = \{z, \text{ such that } z \text{ is in } C, |z| = 1\}$. It is a unique up to isomorphism compact and connected Lie group within the complex plane. It is isomorphic to U(1), the first unitary group. The covering group for T is R, the real numbers with the usual additive group. The covering homomorphism L: $R \rightarrow T$, where $L(r) = e^{2ir\pi} = \cos(2r\pi) + i \sin(2r\pi)$. L is onto but not 1−1. The kernel of the homomorphism is the set of all integers Z, since the inverse image of the identity element 1 in T are integers, Z in R. T is isomorphic to R/Z.#

The process of finding a universal covering group will be described for Lie groups and then followed by an example. Here, it is assumed that there exist several Lie groups sharing a single Lie algebra (Gilmore, 1974). The first thing to find are the fundamental groups associated with each Lie group. One of these groups, G, must be simple, so homotopy $\pi_1(G) = \{I\}$. If more than one of the fundamental groups of a Lie group are simple, then the corresponding Lie groups are isomorphic. Next, find the discrete invariant subgroups D1, D2, . . ., DN also called normal subgroups for the Lie group G. G is the universal covering group for the Lie algebra A. The discrete invariant subgroup D of G is invariant; therefore,

gD = Dg. For all g in G, this could be written as g D g^{-1} = D. Also, they are discrete, that is, usually finite or countable infinite, with no accumulation point. It will be seen next that D, being discrete, strengthens the commutativity property.

The Lie group G always commutes with every element d′ and d″ of D. This commutativity follows by continuously moving g to the identity in $gd'g^{-1} = d''$. Because D is a discrete group, this implies that d′ = d″, because d′and d″ cannot differ continuously; they only differ by delta amounts. This is a direct result of having discrete elements. The group D is the center of G; it is the largest, discrete invariant subgroup of G. All other discrete invariant subgroups Dj are themselves a subgroup of D. Most importantly, all other Lie groups within this family are isomorphic to G/D. The number of distinct discrete invariant subgroups governs the number of distinct Lie groups in a family corresponding to a single Lie algebra. Here, when Di and Dj are isomorphic, but distinct, each might correspond to a distinct group in the family. Moreover, each of the groups G/Di within this family has a fundamental topological group π_1(Gi), which is isomorphic to the corresponding algebraic discrete invariant group Di. This relationship provides a strong link between algebraic and topological structures in Lie groups. The discrete invariant group Di is algebraic, but the fundamental group π_1(Gi) is a group because of homotopy, a topological quality; nevertheless, isomorphisms between these structures control the consequences of the universal covering group.

The next example will involve a family of three Lie groups, all of which have the same Lie algebra h = su(2, C). All three Lie groups are locally similar to each other, but they are not identical.

Example 12.10:

This is probably the simplest instance of the covering space for Lie groups. It involves three distinct Lie groups illustrated in Fig. 12.11. These three groups are given next. They are, first, the Lie group of special orthogonal rotation SO(3). The next is SLC(1, Q), which is the special linear group of one-dimensional quaternions. The last Lie group is the

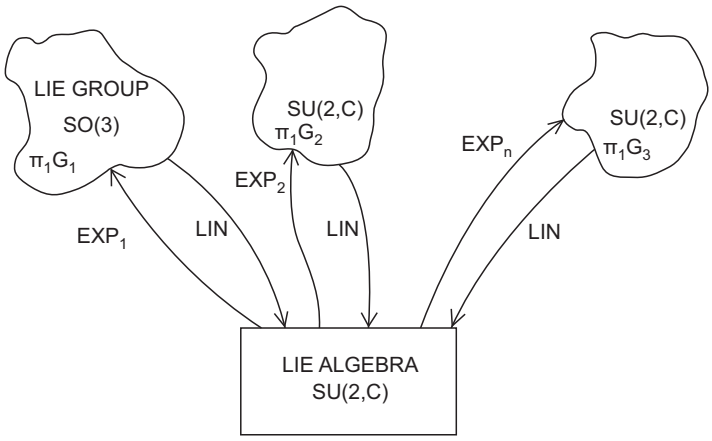


FIGURE 12.11 Universal covering group.

special unitary group SU(2). They all possess the common Lie algebra, su(2), even though, respectively, they involve three-by-three matrices, quaternion scalars, and lastly two-by-two matrices. The Lie group SU(2) is the extension of SO(3) by Z2. All these groups when linearized gave the same Lie algebra. That is, the linearization gives the same Lie bracket and generators. On the other hand, now going the other way. Different representations provided these different Lie groups for the unique Lie algebra su(2) when exponentiated.

Since the fundamental group of SU(2, C) is simple, that is, it only consists of the equivalence class [I]. That is, $\pi_1(SU(2, C)) = [I]$. It follows that SU(2, C) is the universal covering group. Next, the discrete invariant subgroups of SU(2, C) must be determined. Observe that the center of SU(2, C) is D = {I, − I}. It is the largest subgroup of SU(2, C), which commutes with all elements of SU(2, C). This subgroup can be found by utilizing the coordinates in the general matrix structure of SU(2, C) in Example 11.26. Let all entries be zero, except that first let $d = 2\pi$ and then use d = 0. The only other discrete invariant subgroup is {I}. Note that D is isomorphic to $\pi_1(SO(3, R))$. The factor group of SU(2, C) by {I, − I} is isomorphic to SO(3, R). The factor group of SLC(1,Q) by {I} is itself. Also since the discrete invariant subgroup is {I}, the special quaternion group, SLC(1, Q), and the special unitary group, SU(2, C), are isomorphic and homeomorphic.

The quotient space SU(2, C)/{I, − I} consists of an uncountable number of elements, for instance, a, b of SU(2, C) partitioned into cosets. The set of all cosets are of the form {I, − I}, {a, − a}, and {b, − b}, and form an uncountable number. resulting in a new group. Multiplication of cosets [a] x [b] = [a x′ b], where x′ is the multiplication in SU(2, C) and x is the induced multiplication in SO(3, R). SU(2, C) is an extension of SO(3, R) by the fundamental group Z2.#

12.7 The Cornwell mapping

The two-to-one homomorphic map T, from SU(2, C) to SO(3, R) is also referred to as a Cornwell mapping (Cornwell, 1997).

Example 12.11:

For a 2x2 matrix U given below with z, w in C and $|z|^2 + |w|^2 = 1$,

$$\begin{vmatrix} z & w \\ -w^* & z^* \end{vmatrix}.$$

This is a typical matrix in SU(2, C). It is also called a low-dimension spinor representation when written in terms of the Pauli matrices. The image of U is given by T(U), which is:

$$\begin{vmatrix} \mathrm{Re}(z^2 - w^2) & \mathrm{Im}(z^2 - w^2) & -2\mathrm{Re}(zw) \\ \mathrm{Im}(z^2 - w^2) & \mathrm{Re}(z^2 + w^2) & 2\mathrm{Im}(zw) \\ 2\mathrm{Re}(zw^*) & 2\mathrm{Im}(zw^*) & |z|^2 - |w|^2 \end{vmatrix}$$

T(U) is in SO(3, R).#

It is interesting to note that the representation of T(U) in SO(3,R), in the aforementioned example, was used in proving a version of the Solovay-Kitaev theorem for special orthogonal groups. The SU(2,C) version of this theorem is described in a forthcoming section on quantum computing.

References

Cornwell, J., 1997. Group Theory in Physics: An Introduction. Academic Press.
Gilmore, R., 1974. Lie Groups, Lie Algebras, and Some Applications. Courier corp.
Hatcher, A., 2002. Algebraic Topology. Cambridge U. Press.
Humphreys, J., 1972. Introduction to Lie Algebras and Representation Theory. Springer Verlag.
Humphreys, J., 1997. 978−3540900535 2nd ed. Introduction to Lie Algebras and Representation Theory, vol 9. Springer.
Strom, J., 2011. Modern Classical Homotopy Theory, vol 127. American Mathematical Society.
Whitehead, 1978. GTM Elements of Homotopy Theory, Vol. 61. Springer-Verlag.

CHAPTER

13

Spectra for operators

13.1 Spectral classification for bounded operators

This section is a continuation of Sections 8.5 and 8.6. The principal references for a major earlier part of this chapter are Halmos (1957) and He'lein (2014). In Chapter 8, the criteria originating with the first reference (Halmos, 1957) were provided for illustrating the existence of a spectrum, specifically, the spectrum spT in Hilbert space H. The necessary and sufficient condition is repeated here. For an operator T, in B(H,H), spT = {c, such that c in C, and (c I−T) is not invertible iff both 1) and 2) hold}:

1) The image of T is dense in H, and
2) There is a scalar, a> 0 such that for all v in H$\|Tv\| > a\|v\|$.

In the second reference given earlier, criterion (1) is referred to as weakly onto, and the second criterion number (2) is implied to be strongly one to one.

For T, in B(H,H), it was previously shown that the spectrum of T is nonempty and forms a compact set in C. Moreover, the resolvent set, RsT, is an open set in C. This set RsT = {λ, where λ is in C, and such that (T−λI) is one to one and onto, $(T-\lambda I)^{-1}$ is bounded}. The boundedness of the inverse follows also from the bounded inverse theorem. This theorem can be found in Appendix A.6. Additionally, a partition of the spT is provided by three disjoint sets sppT, spcT, and sprT. Briefly, the spectra will be described for T, a bounded operator on a complex Banach space, X differing from {0}. The inner product is not necessary for this development; thus a Banach space is utilized. Numorous examples are provided, and in the unbounded case they are presented in a formal manner. Domain issues are not considered.

1) The point spectrum sppT = {λ in C such that (T−λI) is not one to one on X}.
2) The continuous spectrum spcT = {λ in C such that (T−λI) is one to one on X, but it is not onto X; however, it has ran (T−λI) dense in X}.
3) The residual spectrum sprt = {λ in C such that (T−λI) is one to one and not onto, but with ran (T−λI) not dense in X}.
1) For sppT, Ker(λ I−T) is not {0}; there must exist a nonzero vector f in X such that T f = λ f, f→(T−λI)f. This mapping is not one to one in X; any nonzero multiple of f satisfies this criterion, since any nonzero scalar times an eigenvector is again an eigenvector. In complex n-dimensional space, dim (Ran (T))+dim (Ker (T)) = n. Additionally, f in X is called an

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00013-8

eigenvector, and λ is called an eigenvalue. In finite-dimensional Hilbert space, there only exists this point spectrum. Several examples of the point spectrum have been illustrated using matrix operations in finite-dimensional Banach or Hilbert space. A most trivial illustration of the point spectrum in an infinite-dimensional space is given in Example 8.16. Additional examples in infinite-dimensional Hilbert space with a point spectrum are provided in Examples 13.1 and 13.2.

2) For spcT, Ker(λ I−T) is {0}, and the image of λI−T is dense in H, but Im (λ I−T) is not equal to X. Equivalently, λI−T is dense in H iff ker (λI−T)* = {0}. Since Ker(λ I−T) is {0}, (λ I−T) ($f_1 - f_2$) = 0; this implies that $f_1 = f_2$, so (T−λI) is one to one on X. It is not onto because if it were then Im (λ I−T) = X. Instances of this spectrum are given in Examples 8.17 and 8.18.

3) Ker(λ I−T) is {0}, the image of λI−T is not dense in H, and Im (λ I−T) is not equal to H; then λ is in the residual spectrum. This means that the operator T having a residual spectrum is far away from being onto. The associated spectrum is denoted by sprT. An example of a nonempty residual spectrum is provided in Example 13.3 (Stack Exchange, 2016).

Example 13.1:

Consider, l^p, for p in [1, ∞), with scalars in C. For λ_1, λ_2, λ_3, ... in C and bounded, that is, sup $|\lambda_n| < \infty$. Let T be an infinite scalar matrix T: $l^p \rightarrow l^p$, with T having values of λ in order on the main diagonal. Using the standard basis implies that $Tv_n = \lambda_n v_n$ where the eigenvector is v_n = (0 0 ... 0 1 0...), and the 1 is in the nth tuple position. So sppT is a subset of the set spT. That is, v_n are eigenvectors corresponding to eigenvalues λ_n.

In the closure of the set of all eigenvalues, there could exist a point c in C, and this point is in the spectrum. If there exists a limit point c, for the set of all λ_n, then c cannot be in sppT, but c is in spT. This will happen in the closure of the set of all eigenvalues. For instance, if $\lambda_n = 1/n$, then 0 would be in the spectrum of T, that is, it is sprT or spcT.#

Example 13.2:

Let T: $l^2 \rightarrow l^2$ over the complex field. For any v in l^2, we have that T(v1, v2, v3, ...) = (v1, v2/2, v3/3, ...). The point spectrum is {1, 1/2, 1/3, ...}. This can easily be seen by writing an infinite scalar matrix with values 1/n on the main diagonal. Using the standard basis implies that $Tv_n = \lambda_n v_n$ where the eigenvector is v_n = (0 0 ... 0 1 0...), and the value one is in the nth tuple position. So sppT is a subset of the set spT. That is, v_n are eigenvectors corresponding to eigenvalues λ_n. The value zero is a limit point of eigenvalues. So zero would also be spT.#

Example 13.3:

This is an example of the residual spectrum being nonempty. Let the carrier set for the Hilbert space H be l^2, the space of all square absolute summable sequences with complex entries. For T: $l^2 \rightarrow l^2$, consider T(v1, v2, v3, ...) = (0, v1, v2/2, v3/3, ...). This is a right-shift operation. The Ker(c I - T) = {0}. T v1 = 0, T v2 = v1, T v3 = v2/2 .... So again, the spectrum is not empty, and zero is a spectral point, because of the noninvertible criteria. T

is not onto, since the first tuple in the codomain is fixed. If vn is a unit vector with 1 at position n and 0 elsewhere, then Tvn---> 0. See Stack Exchange (2016). Note that the dual operator T* is such that <T*w, v> = <w, Tv>. Since T(v1, v2, v3, ...) = (0, v1, v2/2, v3/3, ...), substituting into the leftmost inner product gives w1′*v1+w2′*v2+w3′*v3 + ... = w1 • Next expanding the other inner product results in 0+w2 v1+w3 v2/2 + ... Here, wj′* = T*wj. Next, equating coefficients of vj implies that w1′* = w2, w2′* = w3/2, ... Therefore, T* (w1, w2, w3, ...) = (w2, w3/2, ...). It is a left-shift type operation.#

13.2 Spectra for operators on a Banach space

The spectrum of the adjoint operators when they exist is often related to the spectrum of the original operator. The Lumer-Phillips theorem is a useful tool for determining the spectrum of adjoint operators in Hilbert and Banach spaces (Lumer and Philips, 1961). This theorem was used in He'lein (2014), to illustrate the distinct types of spectra for bounded operators in a Banach space. Many of these same examples are repeated in the following section. First, however, Lumer-Phillips theorem specifies that, in a Banach space A with T′ being the dual operator of T in A, the spectra spT = spT′. Moreover, for c in the resolvent set Rs(T), the resolvent operators Rc(T)′ = Rc(T′). Also, for T in a Hilbert space with adjoint T*, it follows that spT* = (spT)* = {c* | c in the spectrum of T, that is, c in spT}. Finally, for c in the resolvent set Rs(T), the resolvent operators: Rc*(T*) = (Rc(T))*.

The Banach space l^1 has a dual space l^∞; additionally, there is a continuous isomorphism from l^∞ into the dual space of l^1. Consider a continuous linear operator ϕ, ϕ: $l^\infty \to C$. Let en, n = 0, 1, 2, ..., in l^1 be a basis where en = (0, 0, ...0, 1, 0, ...) and has one in the nth tuple and zero elsewhere. For v in l^1, where v = (v0, v1, v2, ...), the norm of the difference $\|v - \sum_{n=0}^{N} (vn \cdot en)\|$ is less than or equal to $\sum_{N+1}^{\infty} |vn|$, which goes to zero as N goes to infinity, since vn is absolute summable. Now using ϕ and then employing the continuity condition gives $\phi(v) = \lim_{N\to\infty} \phi[\sum_{n=0}^{N} (vn \cdot en)] = \lim_{N\to\infty} \sum_{n=0}^{N} (vn \cdot \phi[en]) = \sum_{n=0}^{\infty} (vn \cdot \phi[en])$. Since ϕ is continuous for all w in l^1 and since ϕ is bounded, $\|\phi\|$ is less than or equal to c, where c> 0. Also, $|\phi(w)|$ is less than or equal to $\|\phi\|\ \|w\|$. In particular, $|\phi(en)|$ is less than or equal to $\|\phi\|\ \|en\| = c$. So $\phi(en)$ is bounded, and as a consequence, the series for $\phi(v) = \sum_{n=0}^{\infty} (vn \cdot \phi[en])$ is absolutely converging; $\sum_{n=0}^{\infty} |vn \cdot \phi[en]|$ is less than or equal to c $\sum_{n=0}^{\infty} |vn|$. It follows that $\|\phi\|$ is less than $\|\phi(en)\|$ because a bounded operator norm is the infimum of the Lipschitz constant M, where $\|\phi(v)\|$ is less than or equal to $M\|v\|$. This implies that $\|\phi\| = \|\phi(en)\|$.

Several examples that follow (He'lein, 2014) involve the Banach space l^1 and its dual space l^∞. In the following, the range and image of an operator T are used interchangeably. That is, ranT = imT; they are the same. Additionally, the closure of an operator is denoted by closT.

Example 13.4:

Consider the Banach space l^1 of all complex-valued sequences v = (v1, v2, ...), which are absolutely summable. Let L be the left-shift operator L (v) = (v2, v3, ...). The dual operator is denoted by L* and not symbolized as L′ for readability purposes. It operates in the dual space

I^{∞}, the space of bounded sequences w = (w1, w2, ...). Even though the inner product is not applicable, it will be employed as an operational calculus device to quickly find the adjoint. So formally, taking the inner product <w, L (v)> = w1*v2+w2*v3+.... Next, to find L*, use <L* (w), v> and then equate this result to the sum written earlier as <L* (w), v> = w1′*v1+w2′*v2+w3′*v3+..., where the prime, that is, wn′*, indicates the nth tuple of L*(w). Equating coefficients of vk involving <L* (w), v> and <w, L (v)>, and solving for wk′*, gives w1′* = 0, w2′* = w1*, w3′* = w2*, ... This shows that the dual operator L* (w) = (0, w1, w2, ...). The resulting operation is the right shift.

Note that $\|L^n\| = \|(L^*)^n\| = 1$, and the norm equals the spectral radius for both L and L*. Gelfand's formula from Section 8.10, shows this. Here the limit n → ∞, of $\|L^n\|^{1/n} = r(L) = 1 = \text{limit } n \to \infty$, of $\|(L^*)^n\|^{1/n} = r(L^*) = 1$. Because of this result, the spectra of L and L* lie in the closure of an open ball centered at the origin, and of radius one. The ball is denoted clos(O(0,1)). Where clos represents the closure operation. All the points of the open ball are eigenvalues, that is, O(0,1) is a subset of sppL. If v = (1, 0, 0, ...), then v is an eigenvector of L with c = 0 as an eigenvalue, because L(v) = 0 v. Other points c in O(0,1) are also in sppL. To see this, let $v = (1, c, c^2, \ldots)$ then $L(v) = (c, c^2, c^3, \ldots) = c(1, c, \ldots) = c\ v$. Accordingly, the results mentioned earlier show that the open unit ball consists of eigenvalues for L.#

Example 13.5:

Again consider the Banach space l^1, as in Example 13.4. This space consists of all complex-valued sequences v = (x1, x2, ...), which are absolutely summable with L the left-shift operator L (v) = (x2, x3, ...). It was seen in that example that the dual operator L* operates in the dual space l^{∞}, the space of bounded sequences w = (y1, y2, ...). Moreover, the dual operator L* (w) = (0, y1, y2, ...) is the right-shift operator. It was seen that the spectra of L and L* lie in the closure of an open ball centered at the origin with radius one, clos(O(0,1)). The inclusions hold: O(0,1) is contained in sppL, which itself is contained in spL, and this is contained in clos(O(0,1)). Then, since spL is compact, spL = clos(O(0,1)) = spL*, by the Lume-Phillips theorem. In Example 13.4, it is shown that O(0,1) is a subset of sppL. Ultimately, it will be seen that O(0,1) = sppL, that is, the open unit ball is the point spectrum for L.

First note that the boundary of O(0,1) is not in sppL. That is, the boundary points are not eigenvalues. The boundary consists of {c, such that | c | = 1}. Say that c is in ker(c−L) and not equal to {0}. This is proof by contradiction. Let v = (x1, x2, ...) and then (c−L) v = 0. That is, (c x1, c x2, ...) = (x2, x3, ...). A result of this identity is a recursive set of equations. Specifically, the recursion must hold, x(n + 1) = c xn, n = 1, 2,.... Solving, by using backward substitution, gives $v = c\ x1(1, c, c^2, \ldots)$. Since | c | = 1, v is not in I^1, unless x1 is zero. It follows that the boundary is not in sppL, and therefore, the whole point spectrum is sppL = O(0,1).

Next, it will be seen that the point spectrum of L* is empty. Again assume not, that is, there is a complex value c such that (c−L*) w = 0 for a vector w = (y1, y2, ...) in l^{∞}. That is (c y1, c y2, ...) = (0, y1, y2, ...); this also leads to a set of equations, namely: c y1 = 0, c yk = y(k-1), k = 2, 3, ... When c = 0, then w = 0 and cannot be an eigenvector. Otherwise, when c is not zero, then from the equations mentioned earlier, y1 = 0, and then

by recursion, yk = 0 for all k = 2, 3, ... This implies that w = 0, and it cannot be an eigenvector. Thus, sppL* is empty, and all that is known so far is that for the adjoint; the closed ball centered at the origin is the spectrum of L.#

Example 13.6:

Refer to Examples 13.4 and 13.5. The right-shift operator L* operates in the dual space l^{∞}, where for w = (y1, y2, ...), L* (w) = (0, y1, y2, ...). It will be seen that the open ball O(0,1) is in sprL*. For c or c* in O(0,1), let $v = (1, c, c^2, \ldots)$, so v is in l^1, and the same holds for a vector involving the conjugates of c. To be in the residual spectrum of L*, all that needs to be shown is the following: ker (c* I−L*) is {0}, and the image of c*I−L* is not dense in H. These criteria uniquely classify sprL. Consider any vector w in l^{∞}, and using again, only in a formal manner, a map of the form, f: w→<w, v>. Accordingly, f: (c*−L*) w→<(c*−L*) w, v> = < w, (c−L) v> = < w, c v−L v>. Use the definition of the operator L, and notice that c v−Lv = 0, because all points c in the open ball are eigenvalues. This shows ker(c*−L*) = {0}, and the image of c*−L* is not dense, because for any w in l^{∞}, f: (c*−L*) w = 0; therefore, closf(c−L*) cannot be dense in l^{∞}. This shows that the open ball O(0,1) is in sprL*.

Next, it will be shown that the residual spectrum of L* equals the closed unit ball, that is, sprL* = clos(O(0,1)). Employ any complex number c, of unit length and vector u in l^{∞}, and w in C. Use the vector tuples to form the set of equations resulting from the following identity: (c−L*) w = u. Expanding the identity provides c w1 = u1, and (c w2−w1) = u2,..., c wk−w(k-1) = uk, k = 2, 3,.... Solving for vector w: w1 = c*u1, w2 = c* (w1+u2),..., wk = c* (uk+w(k-1)). Back substitution gives the result: $wk = c^*uk + c^{*2}u(k-1) + \ldots + c^{*k}u1.u = (1, c^*, c^{*2}, \ldots)$ is in l^{∞}, because the scalar c is of unit length and u is bounded. Substituting in the tuples of u into the expression for wk gives $wk = c^{*n} + c^{*n} + \ldots + c^{*n} = kc^{*n} \to \infty$ as k→∞ and so the vector w is not in l^{∞}. Thus, $u = \left(1, c^*, c^{*2}, \ldots\right)$ is not in the image of (c−L*). This shows that points on the boundary of the unit circle are not eigenvalues of L*. So far, this shows that the boundary of the unit circle can be in the continuous or residual spectrum of L*. Next, it will be shown that ran(cI−L*) is not dense in l^{∞}. Accordingly, these points will then be shown to be in the residual spectrum of L*.

To see this, it will be shown that O (u, 1/2) intersects im(cI−L*) and is empty in l^{∞}. For v in O (u, 1/2), v = u+w, where the l^{∞} norm, $\|w\|$ is less than 1/2. Consider the equation: (cI−L*) z = v = u+w. Here, z = (z1, z2, ...) is a vector of complex-valued tuples, which is to be the solution of the aforementioned equation. It follows that $c\,z1-0=1+w1, c\,z2-z1=c^*+w2, \ldots, c\,zk-z(k-1)=c^{*k-1}+wk, k=3,4,\ldots.$ Again, by substitution, solving tuple by tuple for the complex vector z yields $zn = c^{*n} + \sum_{j=1}^{n}\left(c^{*n+1-j}wj\right)$. Now using the bounds on c and the vector w shows that |zn−c*n| is less than or equal to $|\sum_{j=1}^{n}\left(c^{*n+1-j}wj\right)|$, which shows that |zn−c*n| < n/2. The vector zn is not bounded and therefore not in l^{∞}. This follows since |zn−c*n| is greater or equal to |c*n| − |zn| = n − |zn|or|zn| is greater than or equal to n−|zn−c*n| > n/2. This shows that the closure of (cI−L*) is not l^{∞}. Thus, sprL* = |c|. Moreover, sprL* is the whole closed unit disk in C.#

Example 13.7:

For the left-shift operator L, in Example 13.4, it will be seen that the residual spectrum is empty. Since it was shown in Example 13.5 that O(0,1) = sppL, only the boundary of the unit ball need to be considered. So use a unit complex value c. That c is in sppL* is false since Example 13.6 shows that sppL* is empty on the boundary $|c| = 1$. The continuous spectra for L and L* are found using the fact that the three types of spectra are mutually disjoint and the union is the total spectra for each operator, which shows that spcL* is empty. Use the fact that sppL* is empty on $|c| = 1$. Also, apply the proposition 6.3 (ii) from He'lein, (2014). It states that if the ran (cI−L) is not dense in a Banach space, then c is in the sppL*. From the contrapositive of this proposition, it follows that if sppL* is empty on $|c| = 1$, then the ran (cI−L) is dense in l^1; this implies that spcL equals the boundary of O(0,1). So, the boundary of the unit ball is the continuous spectrum of L.#

A summary of the results from Examples 13.5–13.8 appears in Fig. 13.1 (He'lein, 2014). In this figure, the closed unit disk appears twice, once to the left, for the left-shift operator L, and once for its dual operator L*. The spectrum of L consists of a point spectrum sppL in the open unit disk in C. On the boundary of this disk is the continuous spectrum, spcL. The adjoint operator indicated on the right-hand side diagram shows that everywhere in the closed disk, L* has a residual spectrum, sprL*.

13.3 Symmetric, self-adjoint, and unbounded operators

In quantum technology, the self-adjoint operator is of principal importance in that it is uniquely utilized in the observation process. This section and the next few sections are used in reviewing, extending, and illustrating numerous facts pertaining to self-adjoint operators both in the bounded and unbounded cases. Several illustrations of self-adjoint operators have been exhibited, mainly in terms of matrix operators. A simple example of a self-adjoint operator on an infinite-dimensional Hilbert space follows.

Example 13.8:

This instance of a self-adjoint operator extends the results given in Example 13.1. Let T: $l^2 \to l^2$ over the complex field. For any v in l^2, use the mapping, Tv = T(v1, v2, v3, ...) = (v1, v2/2, v3/3, ...). To be in l^2 means that the sum $\sum_{n=1}^{\infty} (vn)^2$ converges. Consider w in

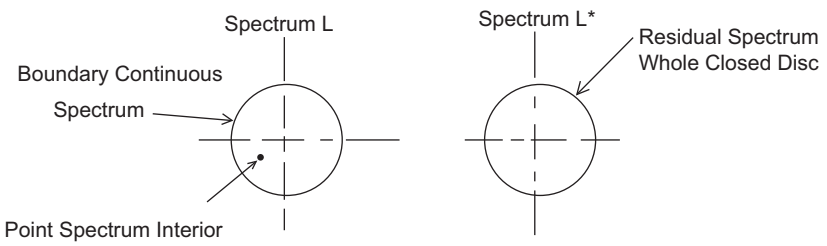


FIGURE 13.1 Spectra for operator L in l^1 and its dual in l^∞.

l^2, where w = (w1, w2,...), and then the inner product <w, Tv> = w1*v1+w2*v2/2+w3*v3/3+.... The definition of the adjoint operator involving the inner product is <T*w, v> = w1′*v1+w2′*v2+w3′*v3 +.... Here, wj′* is T*wj. Then to see if there exists a relation between T and T*, equate coefficients involving vn. This gives w1′* = w1*, w2*′ = w2*/2, w3′* = w3*/3.... As a consequence, T*w1 = w1, T*w2 = w2/2, T*w3 = w3/ 3, ... Accordingly it follows that T* = T, and so the operator T is self-adjoint.#

Example 13.9:

Let T: $l^2 \rightarrow l^2$ over the complex field. For any v = (v0, v1, v2, v3, ...), in l^2 say that Tv = (v1, v2/2, v3/3, ...). To find T*, notice that for w in l^2, <w, Tv> = w0*v1+w1*v2/2+.... Also forming <T*w, v> = w0′*v0+w1′* v1+.... Then equating the two inner product expressions together and then setting the coefficients of vn equal to each other yields: w0′* = 0, w1′* = w0*, w2′* = w1*/2, ... As a consequence, T*w = (0, w0, w1/2, w2/3, ...). So in this example T* does not equal T, so it is not self-adjoint. Notice, also that <T*w, v> = 0+w0*v1+w1*v2/2+.... Accordingly, <T*w, v> does not equal <w, Tv>, and so T and T* are not even symmetric, which is a must in this case.#

In the previous example since the operator T is not self-adjoint, it automatically follows that the operator is not symmetric. The result does not hold for unbounded operators that in fact are the most important in quantum. For a not necessarily bounded operator on Hilbert space H, the domain of T, domT, is a subset of H. The adjoint T* has domain domT*, which is also a subspace of H with vectors w such that there is a vector v in H such that <v, Tu> = <w, u>. By letting T*w = v, define the adjoint operator T*.

The graph of T is graph(T) = {(v, Tv) such that v is in domT}. An operator S extends T means that graph(T) is a subset of graph(S). For a symmetric operator T, the graph(T) is a subset of the graph(T*). The operator T is self-adjoint when graph(T) = graph(T*). In this case, domT = domT* and T = T*. A symmetric operator having domT = dom T* is self-adjoint.

A key theorem for quantum is the Hellinger-Toeplitz theorem stating that when T is defined on a Hilbert space for all v, w in H such that <Tv, w> = <v, T w>, then T must be bounded (Teschl, 2009). Since an everywhere-defined symmetric operator on H is also self-adjoint, then it follows that for a self-adjoint nonbounded operator T, its domain must be a proper subset of H. The operator T in this case must be a partially defined operator on H.

In several areas of quantum, operators are sometimes defined on a subset of a Hilbert, or a Banach space. This most often occurs for the adjoint operation, as well as for the closed operation. First, for the adjoint, T is defined on dom(T), which is a subset of Hilbert space H_1, and T: dom{T}$\rightarrow H_2$. H_2 is another Hilbert space. In this case, T* is defined by T*(u) = v, which means <v, w> = <u, T w>, for all w in the dom(T). In the above inner products, the first inner product is in H_1 and the second is taken in H_2. It is assumed that T* is a closed operator and that dom(T) is dense in H_1. When this is true, <u, T v> = <T* u, v>, for v in dom(T) and u in dom(T*). Similarly, assume that B_1 and B_2 are Banach spaces, and consider T in L (B_1, B_2). T is said to be a closed operator on dom (T), which is a subspace of B1 when given that vn converges to v in dom (T), and given that Tvn converges to w in B_2, then w = Tv.

The operator T is said to be densely defined when dom(T) is dense in B_1. T is fully defined when dom(T) = B_1. Closure of an operator T means that there exists a closed operator S, such that dom(T) is a subset of dom(S), and S = T on dom(T). Also, the graph norm on dom(T) is given for v in dom(T), $\|v\| = \|v\| + \|T\ v\|$; the first norm is taken in dom(T), the second in B_1, and the third in B_2.

For a nonbounded operator T on a Hilbert space H, it is a closed operator when its graph is closed. The graph of T is a subset of the Cartesian product H x H. It is a vector space. The graph of T is Gr(T) = {< v, T v> | v is in dom(T)}. The closed graph theorem says that an everywhere-defined linear map with a closed graph is continuous. See Appendix A.6. A partially defined linear map with a closed graph can be not continuous.

Example 13.10:

Consider the Hilbert space $H = l^2$, that is, all complex-valued absolute sum squared sequences that converge. Let dom(T) be equal to all sequences such that the sum $\sum |n\ vn|^2$ converges. Let T: dom (T) → H by Tv = T (v1, v2, v3, . . .) = (v1, 2v2, 3v3, . . .). The closure of dom(T) = H, so dom(T) is dense in H. This follows since dom(T) contains all sequences that have at most a finitely many nonzero terms. T is not one to one due to the finite number of nonzero terms. Two identical finite vector tuple strings in the range correspond to an infinite number of different terms in the domain whose image is the same finite tuple sequence. Additionally, T is unbounded, but it is self-adjoint, since <T*w, v> = w1′*v1+w2′*v2+. . . and <w, Tv> = w1*v1+w2*2v2 + ... Here wk′* are tuples from T*w. Then equating coefficients of vk gives wk′* = wk/k. This means that T*w = (w1, 2w2, 3w3, . . .). Moreover, the spectrum for T is the point spectrum: sppT = {1, 2, 3, . . .} since Tvn = n vn.#

Example 13.11:

For the unbounded operator T, having an adjoint T* in the Hilbert space H with T: dom (T) → H. Let c* be in sppT*. That is, c* is in the point spectrum of T*. It will be seen that c will be in the union of sppT with sprT, specifically, that is, it will be in one of these. Since c* is an eigenvalue for T*, there is an eigenvector v in dom(T*) such that T*v = c* v. For all w in dom(T), it follows that <v, Tw> = <T*v, w> = <c*v, w> = c <v, w>. So, <v, c w> − <v, Tw> = 0, and because this holds for all w, this means that <v, (cI − T) w> = 0 for all w in dom(T). Consequently, it follows that the ran(cI − T) is not dense, and so the spectrum cannot be continuous.#

Example 13.12:

Even though the position operation Q in $H = L^2(R)$ is unbounded, it is, however, self-adjoint. The inner product for f in H and the product x • g in H, where g is also in H, are given by the integral $\int_{-\infty}^{\infty} f^* \cdot (x \cdot g)\ dx) = <f, x \cdot g>$. It follows that this quantity equals <x • f, g>. The latter inner product is the integral $\int_{-\infty}^{\infty} (x \cdot f^*) \cdot g\ dx) = <x \cdot f, g>$. So Q(f) = x • f, and so the multiplication operator Q is self-adjoint.#

Example 13.13:

The momentum operator is also unbounded, but it too is self-adjoint in $H=L^2(R)$. The inner product in this case is $<f, -id/dx\ g>$; it is the integral $\int_{-\infty}^{\infty} f^* \cdot (-id/dx\ g)\ dx$. Here it is assumed that the derivatives are integrable. Now integrating by parts gives $(f^* \cdot g)$ evaluated between minus infinity to plus infinity, along with the integral $\int_{-\infty}^{\infty} (i\ d/dx\ f^*) \cdot g\ dx$. Assume that the product $(f^* \cdot g)$ goes to zero; then this result in terms of the inner product is $<-id/dx\ f, g>$, thus showing that this operator is also self-adjoint.#

13.4 Bounded operators and numerical range

As mentioned in Section 7.1, the numerical range is very useful in characterizing the types of spectra associated with bounded operators on Hilbert space. To see this, consider the operator T in the Banach space or actually the C* algebra, B(H). The numerical range for an operator T is $\sup_{\|v\|=1} <v, T\ v>$. Assume that $cI-T$ is not invertible, and it is not even one to one. Then, there exists v nonzero in H with $(T-I\ c)\ v=0$. Use this identity, $Tv=c\ v$, and form the inner product with v, $<v, Tv>=c\ <v, v>$, where $\|v\|=1$. This shows that the eigenvalue c is in the numerical range. As a consequence, the point spectrum sppT is a subset of the numerical range.

Now assume that the range of $(c-T)$ is not dense in H; this means that the closure clos $(c-T)$ is a proper subset A of H. Then there exists a unit vector w in $A^{\perp}$, such that the inner product $<w, (T-c)\ w>=0$, that is, $<w, T\ w>=c$. So again c is in the numerical range. Thus, the residual spectrum sprT is part of the numerical range of T.

Lastly, assume that the range of $(T-c)$ is dense in H, but not equal to H. In this situation, there exists a sequence of normalized vectors, w_n in H such that $(T-c)\ w_n \to 0$. A sequence going to zero has to exist. Otherwise, $(T-c)$ would be bounded from below, and as seen in Section 7.6 and earlier in this chapter, dense and bounded from below implies $(T-c)$ is invertible. Using the inner product with the sequence w_n gives $<w_n, (T-c)\ w_n>=<w_n, T\ w_n>-c<w_n, w_n>\to 0$. As a consequence, this time c is in the closure of the numerical range. In this case, the continuous spectrum spcT is a subset of the closure of the numerical range.

A list of properties for the numerical range W(T) for n by n complex -valued matrix are provided as follows (Gustafson and Rao, 1997):

1) W (T) is a compact subset of C.
2) W (T) is a convex set in C, by the Hausdorff-Toeplitz theorem.
3) W (T) contains the spectrum of T.
4) If $T=T^*$, then W (T) is the closed interval on the real line. Moreover, it consists of end points that are two eigenvalues of T.

The next example illustrates the use of eigenvalues for finding the numerical range. It also illustrates Property 4). This same example was solved by numerical simulation in Example 7.3 where a most general qubit in C^2 was inserted into the numerical range formula.

Example 13.14:

The eigenvalues for the self-adjoint matrix T =

$$\begin{vmatrix} 2 & i \\ -i & 1 \end{vmatrix}$$

are found from the solution of the characteristic equation: $t^2 - 3t + 1 = 0$. The eigenvalues are $(3 + -5^{1/2})/2. = .382, 2.618$. The numerical range is W (T) = [.382, 2.618].#

More generally, for an operator T on Hilbert space H, several other properties hold for the numerical range (Shapiro, 2004).

1) W (T) is invariant under unitary similarity.
2) W (T) lies in the closed disk of radius $\|T\|$ centered at the origin.
3) W (T*) = {c* for all c in W (T)}.
4) W (I) = {1}.
5) For a, b in C, and T in B(H), W (a T+b I) = a W (T)+b.
6) If T is nilpotent, that is, $T^{n+1} = 0$, then the spectrum of T is zero in clos W (T).

Example 13.15:

Use as the carrier set the algebra of 2 by 2 complex-valued matrices T. If T is the matrix with all zeros except for the first row second column entry, here, let entry $T_{1,2}$ be equal to one, T =

$$\begin{vmatrix} 0 & 1 \\ 0 & 0 \end{vmatrix}$$

Then sp(T) = 0 and $\|T\| = 1$. Additionally, T is nilpotent order two since $T^2 = 0$.#

For unbounded operators, relations between the spectrum and the numerical range are the following: (StackExchange, 2015).

1) If T is normal, then the whole spectrum of T lies in the clos W (T*).
2) If T is symmetric, contained in closT contained in T*, then the spectrum need not be real, but closW (T) is a subset of R.
3) If T is not a closed operator, and T is a subset of closT = T**, then the spectrum is complex and it contains the closW (T).

For a self-adjoint bounded operator T on H, there is a relationship between the numerical range W(T) and an eigenvalue of T. When a = inf $<$v, Tv$>$ for all $\|v\| = 1$, then a is in spT (Gustafson and Rao, 1997).

As mentioned earlier, the real part of the dissipative operator T in a Hilbert space H is one such that for all v in H, the real part of the numerical range is nonpositive. That is, Re $<$v, Tv$>$ is nonpositive, with $\|v\| = 1$. The next few examples will illustrate this concept.

Example 13.16:

Let the carrier set for the Hilbert space be H=R^n, with the usual inner product if T = −I, that is the negative of the diagonal identity matrix; then for all v in H, $<v, T\,v> = -\|v\|^2$, as such T is dissipative.#

Example 13.17:

Another instance of a dissipating operator T this time in $L^2(R)$ intersects $Co^1(R)$. This is the space consisting of continuously differentiable functions of compact support, which are absolutely square integrable. Let T be a momentum type operator, T = d/dt.

Then <T f, f> = integral, $\int_{t=-\infty}^{\infty}$ f'* f dt, where the prime stands for derivative. Now integrating by parts gives <T f, f> = $\int_{t=-\infty}^{\infty}$ − f* f'dt = − <f, T f> = − <Tf, f> *. So the real part of <T f, f> = 0, because when z = − z*, z is pure imaginary. Accordingly, T is dissipative.#

In a Banach space A, the definition of an operator T being dissipative is extended because there need not be an inner product. In this case, if for all c> 0 and all v in dom (T), $\|(c-T)\ v\|$ is greater than or equal to $c\|v\|$; then T is dissipative. A quite complete compilation of dissipative operators in a Banach space is in Lumer and Philips (1961) and Engel and Nangel (2000). These properties are needed in proving the Lumer-Phillips theorem previously mentioned relating the spectrum of an operator with the spectrum of its dual operator.

Whenever Re <v, Tv> is nonnegative, with $\|v\| = 1$, T is called an accretive operator. Applications to these, as well as dissipative operators, are in creating one-parameter semigroups, much like in Stone's theorem. In the present case, the semigroup consists of operators in a Banach space of the form, e^{tT}, when a dissipative operator is utilized. The corresponding operators in a Banach space are of the form e^{-tT} whenever the generator is an accretive operator T. However, many of these operators are partially defined (Delaubenfels, 1988).

13.5 Self-adjoint operators

As previously mentioned, this and the next section provide a review as well as an extension of concepts related to self-adjoint operators. Of most importance, in quantum disciplines, observables are operators in a C* algebra, A. They are self-adjoint operators T, such that T = T*. Moreover, these operators are measurable and defined via RRT, similar to the procedure of finding the dual space. The method is very similar to the creation of the adjoint T*, for the case where T: H→H was considered previously.

Suppose that T: H1→H2, where H1 and H2 are Hilbert spaces. Additionaly assume for now that T is bounded, that is, T is in B (H1, H2). As in the dual space development, a functional f will be defined. It is such that for fixed w in H2, f: H1→C, where for any v in H1 it follows that f (v) = <w, T v>. The functional is linear and bounded since T is bounded and also |f (v)| is less than or equal to $\|w\|\ \|T\ v\|$. This inequality is a direct result of the CBS inequality. Additionally, the last expression is less than or equal to $\|w\|\ \|T\|\ \|v\|$ since T is bounded. Moreover f is in the dual space of H1.

Using RRT, for every w in H2, there is a unique u in H1 such that f (v) = <u, v>, holding for all v in H1. This defines a mapping: T*: H2→H1, which is called the adjoint of T, mentioned previously in the last section. The mapping T* is given by T* (w) = u. In short, the defining property of T* is <w, T v> = <T* w, v>, for all v in H1 and w in H2. The first inner product mentioned earlier is in H2, and the second is in H1. Since T is assumed to be bounded, it follows that T* is also linear and bounded, that is, T* is in B (H2, H1). In addition, the following holds for T and T*.

1) Norm equality: $\|T^*\| = \|T\|$.
2) Involution: $(T^*)^* = T$.
3) Commuting norm: $\|T\ T^*\| = \|T^*T\| = \|T\|^2$.
4) Inverse: If T is invertible, then so is T* and $(T^{-1})^* = (T^*)^{-1}$.
5) Additive: $(S+T)^* = S^*+T^*$, where S is also in B (H1, H2).
6) Anticommutative: $S^*\ T^* = (T\ S)^*$.

A unique representation involving self-adjoint operators can be made for a bounded operator T. Let Q and S be self-adjoint operators in the C* algebra A; then T can be represented as $T = Q+iS$. Therefore $T^* = Q-iS$. Adding T plus T* and then subtracting T* from T gives $Q = (T+T^*)/2$ and $S = (T-T^*)/(2i)$, respectively. The operators Q and S can be thought to be the real and imaginary parts of T, in that order.

If K is a closed subspace of H, then K is itself a Hilbert space. In this case, any vector v in H can be written as u+w, where $<u, w> = 0$ and u is in K. The space of all those vectors orthogonal to vectors in K is called K perp, that is, write it as perp(K) or $K^{\perp}$. A projection operator P can be associated with this procedure. Indeed, let P: $H \to K$, where P (v) = u. Such an operation is both idempotent and self-adjoint. Note that (I−P) (v) = w. If also $v' = u'+w'$ with u′ in K and w′ in perp(K), then $<P(v), v'> = <u, u'> = <v, u'> = <v, P(v')>$. Furthermore, $<u, w> = <P(v), (I-P)(v)> = <v, P\ (I-P)\ (v)> = 0$. When K is a closed subspace of H, then perp (K) is also closed. An important property of a vector subspace K of a Hilbert space H is that the closure of K equals perp(perp(K)). If K is a closed subspace of H, then again K = perp(perp(K)). Moreover, K is closed iff $K=(K^{\perp})^{\perp}$. To see this, first assume that $K=(K^{\perp})^{\perp}$. Then choose a sequence $v_n \to v$, where v_n is in K and v is in H. It will be shown that v is in K. Take w in $K^{\perp}$, then $<w, v_n> = 0$ for all n, and then using the continuity of an inner product: $\lim_{n\to\infty} <w, v_n> = <w, \lim_{n\to\infty} v_n> = <w, v> = 0$. This shows that v is in $K=(K^{\perp})^{\perp}$, and so K is closed.

Going the other way, if K is a closed subspace of H, then $K=(K^{\perp})^{\perp}$ will be shown. Taking the inner product of v in K and any w in $K^{\perp}$ gives $<v, w> = <w, v>^* = 0$, thus showing v is also in $(K^{\perp})^{\perp}$. So K is a subset of $(K^{\perp})^{\perp}$. To see the opposite set inclusion, let v be in $(K^{\perp})^{\perp}$. Then represent $v = w+z$, where w is in K and z is in $K^{\perp}$. Using linearity, $0 = <v,\ z> = <w,\ z> + <z,\ z> = 0 + <z,\ z> = \|z\|^2$, which implies that $z = 0$. Accordingly, v = w and so $(K^{\perp})^{\perp}$ is a subset of K, and therefore they are equal.

When T is bounded, there is a relation between the kernel and the image of the adjoint operation, namely Ker(T) = perp(Im(T*)) and also that the closure of Im(T*) = perp(ker (T)). Additionally, if K is closed, then perp(K) = perp(closure(K)) = closure(perp(K)). Also, perp(perp(K)) = closure(K). For a closed subspace K, it follows that perp(perp(K)) = K. Moreover, using $w = u+v$, then for P in B(H) and P(w) = u, $\|P(w)\|^2 = \|w\|^2 - \|w-u\|^2$; this follows, because $w-u = v$ and $<u, v> = 0$. So, $\|u\|^2 = \|w-v\|^2 = \|w\|^2 - \|v\|^2$.

Relations between the adjoint and the kernel are as follows:

1) Ker(T*) = perp(ran(T)).
2) Perp(ker(T*)) = clos(ran(T)).

An example will be given using the basis for a momentum operator, the derivative operation. It will illustrate a closed operator on a subspace of a Hilbert space, where this operator is not bounded.

Example 13.18:

Consider the Lebesgue space of square absolute value integrable complex-valued functions on [0, 1]. It is denoted by L_2 [0, 1]. Again, let the unbounded momentum type operator T be employed where T = d /dx. Then, the domain of T is D(T) where D(T) = {f in L_2 [0, 1], such that f is absolutely continuous, where d f /dx is in L_2 [0, 1] with f(0) = 0}. Then T is closed in D(T) (Park, 2013). It is not closed in the Banach space C[0, 1]. That is, it is not closed in the domain {f in C[0, 1], such that d f /dx is in C [0, 1] with f(0) = 0}. See Example 8.1 for details.#

In a finite-dimensional Hilbert space, the adjoint operator is guaranteed to exist by RRT; however, in an infinite-dimensional Hilbert space H, the adjoint may not exist at all. Sometimes it will exist and be of little use. First of all, the adjoint will exist when and only when T is densely defined. In this way, RRT can be used on the inner product showing existence and uniqueness of the adjoint operation. The adjoint is of great use whenever the adjoint itself, T*, is densely defined, and this is when T is closable. The definition of the adjoint is given as follows: For all v in dom(T) and for all u in dom(T*), then the inner products are equal, <u, Tv> = <T*u, v>. Because, the closure of the domain of T = clos (dom(T)) = H, therefore T* will uniquely be found for u in dom(T*).

Consider a vector u in H; if there is a w in H such that <u, Tv> = <w, v> for all v in dom(T), then <u, Tv> is a continuous linear form on dom(T). When <u, Tv> is continuous on dom(T), it can be extended by continuity on all of H. Next, RRT can be applied showing there is a unique w in H such that <u, Tv> = <w, v>. The result follows when and only when dom(T) is dense in H. This shows the adjoint exists and is unique when T is bounded.

13.6 Normal operators and nonbounded operators

The spectrum of a bounded operator is compact and nonempty; therefore, it is closed and bounded in the complex plane. Because of these properties, the spectral radius r (T) can be defined for T in A. It is r (T) = sup {|c|, such thatc is in the spectrum of T}. Additionally, the supremal radius is achieved for some c in sp(T), and also r (T) is less than or equal to$\|T\|$. Moreover, the spectrum can be found using the Gelfand formula: r (T) = limit as n goes to infinity for $\|T^n\|^{1/n}$. When the operator T is normal, it follows that the spectral radius is the norm, that is, r (T) = $\|T\|$. As a consequence, if T is in a C* algebra, then$\|T\|^2 = r(T^* T)$.

Example 13.19:

Because a self-adjoint operator T, H:→H is normal, the spectral radius of T equals its norm that is r (T) = $\|T\|$. So, for instance, if spT = 0, this implies that T = 0. Note that the operator T =

$$\begin{matrix} |0 & 1| \\ |0 & 0| \end{matrix}$$

The spectrum spT = 0, but T is not zero, nor is it a normal operator.#

In general, however, the spectral radius can be smaller than the norm, as seen, for instance, using nilpotent operators.

Previously for a bounded operator T in the C* algebra A, it was seen that T can be represented using two self-adjoint operators Q and S, as $T = Q + iS$. Therefore, $T^* = Q - iS$. Multiplying T by its adjoint T* gives $T\,T^* = Q^2 + i(S\,Q - Q\,S) + S^2$. Next multiplying $T^*\,T = Q^2 + i(-S\,Q + Q\,S) + S^2$. So whenever $S\,Q - Q\,S = -S\,Q + Q\,S$, that is, when $S\,Q = Q\,S$, it follows that T is a normal operator. Another interesting property of a normal operator is given in the following example.

Example 13.20:

If T is a normal operator in a C* algebra, then $\|T^2\| = \|T\|^2$. To see this, use the C* identity $\|T^*\,T\| = \|T\|^2$ from Section 3.8; then it follows that $\|(T^*\,T)^2\| = \|T\|^4$. On the other hand, squaring $\|T^2\|$ gives $\|T^2\|^2$, and applying the C* identity to this quantity leads to the equation $\|T^2\|^2 = \|(T^*)^2\,T^2\|$. Employing the normality condition winds up with $\|T^2\|^2 = \|(T^*\,T)^2\|$. So $\|T\|^4 = \|T^2\|^2$, and taking square roots completes the proof.

A distinct proof uses the spectral radius. As mentioned earlier, let $r = \text{limit}_{n\to\infty}\,\|T^n\|^{1/n}$. For integer n larger than zero, $\|T^{2^n}\|^2 = \|(T^{2^n})^*T^{2^n}\| = \|(T^*T)^{2^n}\| = \|T\|^{2^{n+1}}$. So, by mathematical induction $\|T^{2^n}\| = \|T\|^{2^n}$, for all n larger than zero. By Gelfands formula, $r = \text{limit}_{n\to\infty}\|T^{2^n}\|^{2^{-n}} = \|T\|$.#

For normal operators T, the residual spectrum, that is sprT is nonexistent (Conway, 1990). So the spectrum is comprised of the point and continuous spectrums only. This will be seen again in a later section. Moreover, normal operators have numerical radius the same as spectral radius. There is a close relation between a nonzero normal operator T in a C* algebra and the states within a C* algebra. For every such operator T, there exists a state g where $g(T^*T) = \|T\|^2$ and $|g(T)| = \|T\|$. For every such nonzero normal operator T in A, there exists a pure state g where $|g(T)| = \|T\|$.

The set of all compact operators on H forms a C* algebra. It is nonunital in the infinite-dimensional case. However, an approximate identity can be developed, since this structure is isomorphic to the space of square summable sequences, l^2 (Murphy, 2014).

For two nonbounded operators S and T on H, T is an extension of S whenever dom(S) is a subset of dom(T) and S (v) = T (v) for all v in dom(S). Here, S is said to be a restriction of T. So, S is said to be closable if it has a closed extension T. A core of a closed operator T is a subspace X contained in the dom(T) when T is the closure extending dom(T) intersecting X. The core is also called the essential domain; it is such that for v in dom(T) there exists a sequence vn in X such that $\|v - vn\| + \|Tv - Tvn\| \to 0$ as $n \to \infty$ (Reed and Simon, 1980).

For any nonbounded operator T on H that is also closable, the closure of T denoted by closT is defined to be the smallest closed extension. It is given by the intersection of all closed sets containing T, and additionally, closT is unique. For T densely defined on dom (T) in H, T is called symmetric, which means that $<u, Tv> = <Tu, v>$ for all u and v in dom(T). A symmetric densely defined unbounded operator T has closure: $\text{closT} = T^{**}$. Nonsymmetric operators need not be closable. It is easy to use a test for the self-adjoint criteria of a closed symmetric operator T. It is that the intersection of the resolvent set of T and R is nonempty (Woozy, 2017).

An operator T on H is symmetric, which means that $<T\,x, y> = <x, T\,y>$ for all x and y in the dom(T). This means that dom(T) is contained in the dom(T*). A symmetric operator is such that T is a subset of T**, which itself is a subset of T*. The operator T is symmetric also iff the numerical range of T is real, that is, W (T) is a subset of R. When T is bounded and if for all v, w in H $<v, T\,w>$ is real, then T is symmetric. For closed symmetric operators, the following holds: T = T** is a subset of T*. A rigorous treatment involving all important domain issues can be found in Hall (2013).

A closable operator T is said to be essentially self-adjoint whenever it has self-adjoint closure. It is essentially self-adjoint iff it is symmetric, densely defined and there exists a nonreal complex number c, such that c−T and c*−T have dense range. For a bounded self-adjoint operator T, with smallest spectral value a and largest spectral value b, it follows that $a = -\|T\|$, or $b = \|T\|$. Moreover, the closure of the numerical range = [a, b]. Additionally, the self-adjoint operator T is positive when W (T) or spT lies in the nonnegative real line. The set of all positive operators in a C* algebra A is a convex cone in a real vector space V = {T in A, such that T = T* | .

For a self-adjoint operator T, on Hilbert space H, it follows that ker(T + − i) = {0}. In order to see this, first assume that Tv = iv. Then $<v, Tv> = <v, iv> = i\,<v, v>$, and also $<Tv, v> = <iv, v> = -i\,<v, v>$, but since T is self-adjoint, it follows that ker(T + − i) = {0}. The same is true for T*, that is, ker(T* + − i) = {0}. When T is symmetric and closed with ker (T + − i) = {0}, then ran(T + − i) = H. It will be shown that ran(T + − i) is dense in H. For v in perp(ran(T + i)) and for all w in dom(T). it follows that $<v, (T + i)w> = 0$. So, in this case, $<v, Tw> = < -iv, w>$. For v in the dom(T*) and (T*+i)v = 0, therefore v is in ker(T* + i). Also notice that $\|(T+i)v\|^2 = <(T+i)v, (T+i)v> = \|Tv\|^2 + \|v\|^2$. Consequently, if (T + i)vn is a CS, then it follows that Tvn is a CS and also vn is a CS. When vn converges to v since T is closed, v is in dom(T) and Tvn → Tv.

Unbounded operators should be closed and densely defined. As mentioned previously, the closed graph theorem and the concepts associated with this construct play an important role in determining closure of operators. As an instance of the closed graph theorem determining closure, consider the example.

Example 13.21:

For the densely defined self-adjoint operator T = T*, it follows that T is closed. Begin with the identity $<T^* u, v> = <u, T\,v>$. In terms of the relation, (u, w) is in the graph of T*, which means that $<w, v> = <u, Tv>$ for all v in dom(T). By using continuity $w \to <w, v>$ and $u \to <u, Tv>$ shows that the set of all (u, w) in T* is closed. Therefore, T* is closed, but T = T* and so T is also closed.#

A densely defined symmetric operator T with dom(T) a subset of H → H is self-adjoint iff the ran(T+iI) = ran(T−iI) = H. First notice that to be symmetric means that dom(T) is contained in dom(T*). Assume that the identities involving the range equaling H hold. Next, it is shown that dom(T*) is contained in dom(T). So let w be in dom(T*). We must show that w is also in dom(T). Then there exists an element v in H such that for any u in dom(T), $<w, (T + i)u> = <v, u>$. Now because ran(T − i) = H it follows that v = (T − i)z for some z in dom(T). Because T is symmetric and u is also in dom(T), it follows that $<w, (T + i)u> = <v, u> = <(T - i)z, u> = <Tz, u> + i<z, u> = <z, Tu> + <z, iu> = <z, (T + i)u>$,

but ran(T + i) = H; this implies that w = z since z is in dom(T); then w is in dom(T). So dom (T) = dom(T*).#

Symmetric densely defined operator T, on Hilbert space H, can be self-adjoint whenever T + −i is injective; then T* + −iI becomes an extension, i.e., a bijection from dom(T) onto H. Here T = T*. The problem for T being only symmetric and not self-adjoint is that T + −iI is not necessarily surjective. There exist self-adjoint extensions for T whenever the deficiency indices of T are equal, and only then. These indices are given by dim(ran(perp (T + iI))) and dim(ran(perp(T − iI))) and are mentioned again in the context of partial isometrics. Partial isometries are linear maps between Hilbert spaces. It is an isometry on the orthogonal complement of its kernel. Whenever the indices are the same, these extensions can be parametrized by using linear isometrics or by the use of the Cayley transformation mentioned earlier (Hall, 2013).

For a symmetric operator T, there are relationships between X = dim(perp(T+i)) and Y = dim(perp(T−i)) and unique self-adjoint or essentially self-adjoint extensions of T. When X = Y = 0, these extensions exist. When X = Y is greater than or equal to one, then T has infinitely many self-adjoint extensions. When X differs from Y, then T has no self-adjoint extensions. In any case, whenever T is essentially self-adjoint, then the closure of T, clos(T), is the unique self-adjoint extension of T. This can be seen because if S = S* then since a self-adjoint operator is always closed, it follows that dom(S*) is a subset of dom (clos(T)), which is a subset of dom(S). Since S is self-adjoint, all three domains are equal and so S = clos(T).

The square root of a positive operator P is a self-adjoint operator T, such that $P = T^2$. For any bounded operator P, there is a unique positive square root, it is denoted |T|, and it is the positive square root of T*T. Also the polar decomposition can be performed involving a unique partial isometry U and |T|. Here, T = U|T|, and ker(T*) is a subset of Ker(U). The same results hold for unbounded operators provided that they are closed and densely defined. In this case, |T| could be unbounded. It is a nonnegative self-adjoint operator with the same domain as T. Also U is a partial isometry such that U = 0 on perp(Im(T)). In finite dimensions, however, U can be extended to a unitary operator (Wells and Williams, 2012). In infinite dimensions, it is not true that a partial isometry can be extended to a unitary operation. For instance, the left-shift operator L: $l^2 \to \{(x_2, x_3, \ldots)$ in $l^2\}$ is an isometry on the subspace S = {x in l^2, where $x_1 = 0$}. So L:S→l^2 and is onto.

A simple result for bounded, self-adjoint operators T extends eigenvalue and eigenvector consequences of self-adjoint or Hermitian matrices. Indeed, let a and b be in the point spectrum of T, sppT. The associated eigenvectors v and w are nonzero and such that Tv = av and Tw = bw. When a is different from b; then v is orthogonal to w, since <T v, w> = a <v, w> = <v, T w> = b <v, w>, and (a−b) <v, w> = 0, but a−b is not zero. Thus, <v, w> = 0. For any self-adjoint operator, it follows that T = T* = T**. Additionally, a closed symmetric operator is self-adjoint iff T* is symmetric.

13.7 Spectral decomposition

The spectral decomposition for bounded self-adjoint operators T on the complex Hilbert space H will be described next. These operators are fundamental in quantum since all

observables are of this form. For continuous functions f and g, defined on the spectrum of T, and a and b real scalars, the following identities involving T must uphold:

1) Linear: (a f+b g) (T) = a f (T)+b g (T).
2) Product: (f g) (T) = f (T) g (T).
3) Identity: I (T) = I.
4) Adjoint: f* (T) = f (T).
5) Spectral Invariance: For eigenvalue c of T and v in H, if T v = c v, then this implies that f (T) v = f (c) v.
6) Spectrum: The spectrum of f (T) = {f (c) is such that c is in the spectrum of T}.
7) Positive: f greater or equal to zero implies f (T) is greater or equal to zero.

Example 13.22:

This is another example involving the residual spectrum for a bounded operator T, in Hilbert space H. It makes for an interesting example of an eigenvector. In this case, if c is in sprT, then the conjugate value c* is in sppT*. First, since c is in the residual spectrum of T, then ran(cI−T) is not dense in nor equal to H; it is a subspace. Then, there has to exist a nonzero vector v in H, which is not in clos(cI−T). This element is in the null space of (c*I−T*) and is such that for arbitrary w in H, <v, (cI−T) w> = <(c*I−T*) v, w> = 0. Since w is arbitrary, (c*I−T*) v = 0, implying that c* is an eigenvalue for T* and so c* is in sppT*.#

Example 13.23:

Let T:$l^2 \to l^2$ where T(v0, v1, v2, …) = (v0, 0, v1, 0, v2, …). The operator T produces a stretching type operation over the complex field. This operation is also called dilation or stride. The spectrum of T is the closed unit disk in C. To see this, first consider the adjoint operator. Observe what it does to the kth basis vector vk, where vk = (0, 0, 0, …0, 1, 0, 0, …) with 1 at the kth tuple. So for any w in l^2 by definition of adjoint: <T*vk, w> = <vk, T w> = T(wk) = 0 for all k odd, and for k even, it follows that T(wk) = w(k/2). Likewise, T*vk = 0 for k odd and equal to v(k/2) for k even.

To determine what the adjoint operator is, use w in l^2. Take <w, Tv> = w0*v0+w1* 0+w2*v1+w3* 0+w4*v2+…. This expression equals <T*w, v> = w′0*v0+w′1*v1+w′2* v2+w′3*v3+w′4* 0+w′5*v2+…. Equate tuple coefficients of vk, for k = 0, 1, 2, …. This means that T*(v0, v1, v2, …) = (v0, v2, v4, …). Any c such that |c| <1 is an eigenvalue of T*. In order to see this, use v(c) = sum, $\sum_{k=0}^{\infty} c^k v2^k$; then T*v(c) = T*v1+the sum, $\sum_{k=1}^{\infty} c^k T^* v2^k$. First, T*v1 = 0, since 1 is odd. T*v(c) = sum, $\sum_{k=1}^{\infty} c^k v2^{k-1}$) = cv(c). It follows that c is an eigenvalue of T*. Consequently, for every c such that |c| <1, this value is an eigenvalue. But, since the spectrum is always closed and since $\|T^*\| = 1$, therefore sp(T*) = sp(T) = closed unit disk.#

For any closed operator T on H, the spectrum of T* is the set {c*, such that c is in spT}; also the resolvent operator at c for operator T* is such that Rz(c*) = Rz (c)*. Where the resolvent operator for T at c is Rz (c). This result is similar to the Lumer-Phillips result.

13.8 Spectra for self-adjoint, normal, and compact operators

It was seen that for a self-adjoint operator in B(H), there is no residual spectrum, and moreover, the spectrum is always real. Approximate eigenvalues of an operator in B(H) are such that for any $\epsilon > 0$, there exists a unit vector v in H, such that $\|Tv - \lambda v\| < \epsilon$. Replacing v by $w / \|w\|$ for w not zero, the equivalent condition is as follows: For any $\epsilon > 0$, there exists a nonzero vector w in H, such that $\|Tw - \lambda w\| < \epsilon \|w\|$; then λ is an approximate eigenvalue for T. The set of all such λ for T is called the approximate spectrum of T. It will be denoted by A(T). If T has an eigenvalue, then it is in A(T) since $Tv - \lambda v$ is identically zero. Additionally, A(T) is a true spectrum since A(T) is a subset of spT. The result follows by contradiction. If λ is not in spT, then $T - \lambda I$ is invertible. This implies that $\|v\| = \|(T - \lambda I)^{-1} (T - \lambda I) v\|$ is less than or equal to $\|(T - \lambda I)^{-1}\| \|Tv - \lambda v\|$ for every vector v, that is, $\|Tv - \lambda v\|$ is greater than or equal to $1/\|(T - \lambda I)^{-1}\| \|v\|$. Letting $\epsilon = 1/\|(T - \lambda I)^{-1}\|$ shows that $\|Tv - \lambda v\|$ is bounded below, and therefore λ is not in A(T), which is a contradiction. So the conclusion is that A(T) is always a subset of spT.

For bounded normal operators, even more is true besides the result mentioned earlier; here, spT = A(T), this was mentioned previously. That is, for a normal operator T, all eigenvalues are approximate. What must be shown is that spT is a subset of A(T). Again contradiction will be employed. First say that there is an eigenvalue λ in spT, which is not in A(T). Then for some $\epsilon > 0$, this eigenvalue is such that $\|Tw - \lambda w\|$ is greater or equal to $\epsilon \|w\|$ for all v nonzero. Since T is normal, it follows that $\|T^*w - \lambda^* w\|$ is greater or equal to $\epsilon \|w\|$ for all v nonzero. So both $\|Tw - \lambda w\|$ and $\|T^*w - \lambda^* w\|$ are bounded below, see Section 13.1. If it can be shown that their ranges are dense, then a contradiction occurs, because then, in particular, $Tw - \lambda w$ would be invertible. Instead of showing the range to be dense, it is equivalent, showing that the orthogonal complement of the range is {0}. This follows since $ran(u)^{\perp} = ker(u^*)$, or $ker(u^*)^{\perp} = clos(ran(u))$. So if v isorthogonal to $Tw - \lambda w$, that is, $\langle v, Tw - \lambda w \rangle = 0$ for all w, using the adjoint, this is the same as $\langle T^*v - \lambda^* v, w \rangle = 0$ for all w. The latter equation implies that $T^*w - \lambda^* w = 0$. However, using the previous bounded below property $\|T^*w - \lambda^* w\|$ is greater than or equal to $\epsilon \|w\|$ shows that $w = 0$.

For normal operators, it is easy to show the relationship between eigenvalues of T and of T*. The example below illustrates this fact

Example 13.24:

For T, a normal operator in B(H), if $Tv = \lambda v$ for nonzero v in H, then $T^*v = \lambda^* v$. This follows by letting A consist of all eigenvectors of T for which λ is an eigenvalue. Then since $T(T^*v) = T^*(Tv) = T^*(\lambda v) = \lambda(T^*v)$, it follows that T^*v is in A. Consider the inner product, $\langle u, T^*v - \lambda^* v \rangle$ with the assumption that u is also in A. It follows that $\langle u, T^*v - \lambda^* v \rangle = \langle u, T^*v \rangle - \langle u, \lambda^* v \rangle = \langle T\,u, v \rangle - \lambda^* \langle u, v \rangle = \langle T\,u, v \rangle - \langle \lambda\,u, v \rangle = \langle T\,u - \lambda\,u, v \rangle = 0$. Since T^*v is in A, as well as any nonzero constant times v, it shows that $T^*v - \lambda^* v$ is in A. In turn, this implies that $T^*v = \lambda^* v$.#

For compact self-adjoint operators T, on a Hilbert space H, the spectrum is a real-valued point spectrum with the possible exception of zero. The point zero might be an

accumulation point, and therefore it might not be in the point spectrum. In this case, it is in the continuous spectrum. Again, the operator T can be represented as a converging sum of projections onto the eigenspace {an, such that Tv = an, for v nonzero in H}. That is, Pn is the projection onto this eigenspace and an, and T is the sum, $\sum$ (an Pn). Also, for any w in H, it follows that Tw is equal to the sum, $\sum$ (an Pn(w)).

Example 13.25:

Consider a compact self-adjoint operator T on a Hilbert space H. It is possible that such an operator only has a point spectrum with eigenvalue 0. Accordingly, in this situation, 0 is not a point of accumulation; it is an isolated point. By the projection theorem for normal compact operators, perp(ker(T)) cannot be infinite. For if it had an infinite ON basis {ek} with Tek = ck ek, and ck → 0, then 0 is not isolated (Conway, 1990). Therefore, perp(ker(T)) is finite and so Ker(T) = 0 has infinite dimension. It also follows that there exist countable infinite vectors, all with 0 as their eigenvalue.#

If T and S are compact operators on H, the product TS and ST are compact, as well as the adjoint S* and T*. Moreover, T*T is self-adjoint and positive; as such, it has nonnegative eigenvalues an. A compact operator T is said to be of trace class whenever the one norm of T is finite. The one norm is $\|T\|$ and is such that the summation of the square root of each an, the eigenvalues of T*T, is finite.

For a trace class operator T, the trace is defined as before: Tr (T) is the sum of the inner products: <en, T en>, where en is an ON basis on H. Additionally, the series must converge absolutely and independently of the ON basis employed. When T is also self-adjoint, then Tr (T) = sum of the real eigenvalues of T. Also, for a unitary operator U on H, it follows that the trace: Tr(T) = Tr(U^{-1}T U). See Section 7.5.

13.9 Pure states and density functions

The most important trace class operator previously given is the pure state Rz. It is also called a density operator, because it acts like a probability density function when forming expected values. As such, it is nonnegative and Tr (Rz) = 1. The eigenvalues an of Rz are in (0, 1]. The projection Pn onto the eigenspace is Rz, and it equals the sum $\sum$ (an Pn) where the sum of all the an equals one.

A state is defined to be a linear functional g on a C* algebra A. That is, g: A → C. It must also satisfy:

1) Nonnegative: g (T* T) is greater or equal to zero for all T in A.
2) Identity: g (I) = 1.

It can be shown that if g is linear functional on A, where g (I) = $\|g\|$ = 1, then g is a state. That is, the nonnegativity can be obtained from the norm of g equaling one (Murphy, 2014).

Some useful facts concerning any nonnegative linear functional g, on a C* algebra A will be listed. Here, T and S are operators in A, often called observables, and I is the identity operator in A:

1) CBS Type Inequality: $|g(T^*S)|^2$ is less than or equal to g (T* T) g (S* S).
2) Conjugate: g (S*) = g (S)*.
3) One Law: g (I) = $\|g\|$.
4) Norm Bound: g (T* S* S T) is less than or equal to $\|S\|^2 g(T^*T)$ (Murphy, 2014).

The norm bound is a type of CBS inequality. Pure states were described previously in several different ways. One of them bears a ray involving a convex set within a vector space. Recall that a convex set S is such that for any two vectors u and v in S the line segment joining these two vectors lies wholly in S also. That is, a v+(1−a) u is in S for all a in [0, 1]. An extreme point occurs when a v+(1−a)u = w and u = v = w for all a in (0, 1). So an extreme point on a convex line segment is an end point of that segment. In a C* algebra, the set of extreme points are called pure states and belong to a space of bounded operators over a Hilbert space. An example will be given.

Example 13.26:

The trace on a C* algebra A is a positive linear functional f: A→C such that for all T and S in A:

1) The trace property holds f (TS) = f (ST).
2) Positive means f (T* T) is greater than or equal to zero for T in A. Moreover, whenever f (T* T) = 0 only when T = 0, the trace is said to be faithful. Faithful means for a positive map g, if S is greater or equal to zero and g (S) = 0, then it must be that S = 0.

The unital associative algebra of all n by n complex-valued matrices is a C* algebra, and the trace is the sum of the diagonal elements, Tr(|a><b|) = <b, a>. So, for instance, in C^2, if a = (2 i)′ and b = (3 4i)′, then <b, a> = 10. The outer product, T = a><b, is given below with T*T the conjugate symmetric matrix with positive trace:

$$\begin{matrix} |100 & -80i| & |45 & -60i| \\ |80i & 80| & |60i & 80|.\# \end{matrix}$$

13.10 Spectrum and resolvent set

The analytic functional calculus will be investigated on the spectrum and resolvent set for an operator T, in the Banach algebra B. Recall that the resolvent set RsT is the complement in C of the spectrum of T, spT. For any complex variable z in RsT, the operator (z−T) is invertible. It defines the resolvent operator of T, denoted by $Rz(T)=(z-T)^{-1}$. Whenever |z| is greater than the spectral radius, r(T) of T, a Laurent expansion can be made for $(z-T)^{-1} = z^{-1}(1-T/z)^{-1}$ = sum, $\sum_{k=0}^{\infty} z^{-1}T^k z^{-k}$. The resolvent operator is analytic for |z| > r(T).

The resolvent formula for T is given for the point z as $Rz(T)=(z-T)^{-1}$. Therefore, the identity element I = (z−T) Rz(T) = Rz(T) (z−T). For point w, it also follows that I = (w−T) Rw(T) = Rw(T) (w−T). Accordingly, it can be written that Rz(T)−Rw(T) = Rz(T) (w−T) Rw(T)−Rz(T) (z−T) Rw(T) = Rz(T) (w−T)−(z−T) Rw(T) = Rz(T) (w−z) Rw(T) = (w − z)

Rz(T) Rw(T). Therefore, Rz(T)−Rw(T) = (w − z) Rz(T) Rw(T). By exactly the same argument, it follows that Rw(T)−Rz(T) = (z − w) Rw(T) Rz(T). From these two identities, the conclusion is that Rz(T) Rw(T) = Rw(T) Rz(T).

Now, the complex polynomials p(z) will be considered again. This time, Cauchy's integral theorem (CIT) will be employed. This integral as usual will utilize a closed path within an open set. It must enclose the spectrum spT, of the operator T, from a Banach algebra. The resolvent set is open, and spT is nonempty and compact. Since the resolvent operator is analytic, the product p(z) Rz(T) = p(z)/(z − T) is analytic. Moreover, the closed path of integration is in RsT. CIT gives $p(T) = 1/(2\pi\ i) \int (p(z)/(z\text{-}T))\ dz$. All integrals are over a closed path. For a single contour integration, the orientation is counterclockwise in the finite z plane. With these assumptions, bounds can be given as follows: $\|p\ (T)\|$ is less than or equal to $(1/2\pi)\ \max|p|\sup\|(T-z)^{-1})\| \int d|z|$.

The same procedure as mentioned earlier can be performed using an entire function. All assumptions as in the polynomial case must hold. Since the path of integration is a closed, rectifiable, oriented curve, the maximum of the polynomial or any analytic function for that matter will be on the boundary. As a consequence, the same type of bound as mentioned earlier holds when any entire function is employed. In general, meromorphic functions cannot be handled just yet. However, for any rational function, since there are at most a finite number of poles, the procedure is just as before. In this case, the zeros of the denominator must be enclosed within their own closed path of integration, and a spectral decomposition results. Partial fraction expansions and the use of CIT or residue theory will provide the solution. Again, commutativity of resolvents is subtly employed, that is, Rz(T) Rw(T) = Rw(T) Rz(T).

For an analytic function f(z) on the neighborhood of the spectrum spT, it follows that f(spT) = spf(T), where T is bounded in Banach space X. This is called the spectral mapping theorem, and it is based on the Riesz-Dunford calculus. Numerous extensions to this theorem are given in Haase (2005); here, invertible operators that are not bounded are considered along with different regions of the complex plane.

13.11 Spectrum for nonbounded operators

Nonbounded functions might have their spectrum include the point at infinity. Here, we view the complex plane as extended as on the Riemann sphere. This sphere is geometrically given by stereographic projection and topologically given by the one point compactification of a locally compact Hausdorff space. Moreover, it is a complex one-dimensional manifold. In any case, CIT and other closed path integrals can be used enclosing the point at infinity. The path of integration must have its orientation reversed in this situation. The integration path is always counterclockwise enclosing singularities.

To proceed, let w = 1/z. In the following, leave out the constant $1/(2\pi\ i)$ for the moment, and remember all integrals are closed. Beginning with CIT, CIT = integral, $\int (f(w)\ /\ (w - a)\ dw)$. Then, the integrator is given as $dw = -1/z^2 dz$. The integrand in CIT: f(w) / (w−a) = f (1/z)/ (1/z−a). Next partial fractions are employed on the new integrand: $f(1/z)/(z-a\ z^2) = f(1/z)/z + z\ f(1/z)/(1-za) = f((1/z)/z + f(1/z)/(a^{-1} - z)$. This must be integrated with the opposite orientation. So a minus sign must be included; that is, the

result is $-$f(evaluated at infinity)$=1/(2\pi i)\int f(1/z)/(z-a^{-1})$. The latter integral will either be 0 via Cauchy formula or it will equal f(a) via CIT. In other words, the value of the integral depends on whether a^{-1} is outside the path of integration or not.

The spectrum of unbounded operators will be considered. First recall, for bounded operators, that in a Banach algebra the resolvent set is open. Also, the resolvent is analytic. The spectrum is always nonempty and compact. For unbounded closed operators, the spectrum will be closed. However, the spectrum could, in this case, be all of C or the empty set. Recall that the empty set and the whole set is both open and closed. As in the bounded situation, the resolvent formula for closed operators commutes. For unbounded operators, the point at infinity (the north pole on the Riemann sphere) could be in the spectrum. Thus, also the point infinity can be an analytic point.

Example 13.27:

This is an illustration of an operator T with empty spectrum. The operator was illustrated in Example 13.18. It is densely defined on $L^2[0, 1]$ and is a closed operator. Let $T = d/dt$ where domain T = {f absolutely continuous. such that f′ in $L^2[0,1]$, and f (0) = 0}. To prove that spT = empty set, it will be shown that the resolvent set is the whole complex plane, that is, RsT = C. For g in $L^2[0,1]$, the solution of the differential equation: $f'-\lambda f = g$, f (0) = 0 is found by variation of parameters to be: $f(t) = \int_0^t e^{\lambda(t-x)}g(x)dx$. The resolvent exists for all λ in C, because $RsT(g)=(T-\lambda I)^{-1}(g)=\int_0^t e^{\lambda(t-x)}g(x)dx$. This can be checked: $g=(d/dt-\lambda)\int_0^t e^{\lambda(t-x)}g(x)dx = e^{\lambda(t-t)}g(t)+\lambda\int_0^t e^{\lambda(t-x)}g(x)dx-\lambda\int_0^t e^{\lambda(t-x)}g(x)dx = g(t)$. For f in the domain of T, $dom(T), f(t)=\int_0^t e^{\lambda(t-x)}g(x)\,dx$ corresponds to a bounded operator on $L^2[0,1]$ for all λ in C.#

Example 13.28:

Let T be a normal, possibly unbounded operator over Hilbert space H. Then spT is nonempty. Assuming the contrary, then T^{-1} is a bounded operator, with spectrum sp T^{-1} that can only at most be {0}. This follows using λ not equal to zero in $(T^{-1}-\lambda I)=(I-\lambda T)T^{-1}=\lambda(I/\lambda I-T)T^{-1}=X$. The operator X has a bounded inverse: $X^{-1}=1/\lambda T(1/\lambda I-T)^{-1}$. Bounded normal operators have identical norms and spectral radius, implying that $T^{-1}=0$, yielding a contradiction. If T is unbounded with empty spectrum, then T^{-1} is bounded with zero in the resolvent set.#

13.12 Brief descriptions of spectral measures and spectral theorems

There are numerous versions of the spectral theorem. To begin with a most powerful version, Haase (2018) called the multiplicative version for a normal operator T on a Hilbert space H. In this case, T is unitarily equivalent to a multiplication operator on some L^2 space in a semifinite measure space. Semifinite measures are described in Appendix A.2. As a quick review, a measure is said to be semifinite if for each E in a measure space,

M with measure, μ (E) = ∞, there exists a measurable set, F a subset of E such that $0 < \mu(F) < \infty$. A sigma finite measure is always a semifinite measure.

The spectral theorem for a normal operator T, in Hilbert space H, says there exists a unitary operator U, such that $T{:}H \to L^2(S), T = U^{-1}M\,U$. Here, S is a semifinite measure space that could be a Radon measure in a locally compact space. M represents a continuous multiplication operation in $L^2(S)$. This spectral theorem shows that normal operators in Hilbert spaces have a multiplicative representation in L^2; however, they are not unique. For a self-adjoint operator T, there is a real-valued measurable function f, corresponding to multiplication by a unitary operator: $U{:}H \to L^2$, involving measure space M, where UTU^{-1}. This multiplication operator involves f whose domain is {g in L^2, such that f g in L^2}. When H is separable, as assumed unless stated otherwise, a finite measure space can be employed.

A simpler version of the spectral theorem will briefly be described for a bounded self-adjoint operator T, T: H → H, where H is a Hilbert space over C. Here, T is given by a projective-valued measure, PVM. $T = \int \lambda dE_\lambda$. In this case, the integral is understood as a Stieltjes integral (Apostle, 1974; Rudin, 1987), here, E_λ is called the integrator and is a projection onto the null space consisting of the positive part of $T_\lambda{}^+ = 1/2\{(T-\lambda I)+[(T-\lambda I)^*(T-\lambda I)]^{1/2}\}$. Since T is self-adjoint, the positive square root of the square $(T-\lambda I)^2$ could appear, in place of the bracketed portion. Finally, the limits of integration are from the lower limit, equaling the minimum of the smallest eigenvalue m, or infimum of the continuous spectrum to the upper limit, the larger of the biggest eigenvalue, M, and the supremum of the continuous spectrum. Additionally, using functional calculus integrals of the form $f(T) = \int f(\lambda) dE_\lambda$ will be considered in Chapter 19.

Example 13.29:

The purpose of this example is to illustrate Stieltjes integration in R. The integrand in this example is continuous, for instance, $f(x) = x^3$. The integrator F(x) has a discontinuity at x = 1, and it equals x^2 on [0, 1) and equals 2x on [1, 2]. The integral $L = \int_{x=0}^{2} x^3 dF(x) = \int_{x=0}^{1} 2x^4 dx + 1^3 + \int_{x=1}^{2} 2x^3 dx$. The central term involves the saltus or amount of jump from left hand limits to right hand limits at t = 1. Also the Riemann integrals were obtained using dF(x) = 2 x dx, on [0, 1). And dF(x) = 2 on [1, 2]. A delta function could have been utilized instead of a Stieltjes integral. In any case, continuing gives $L = 2/5 + 1 + 8 - 1/2$.#

References

Apostle, T., 1974. Mathematical Analysis, 2ed. Pearson Pub Co.

Conway, J., 1990. 978-0-387-97245-9 A Course in Functional Analysis, vol. 96. Springer-Verlag.

Delaubenfels, R., 1988. Totally accretive operators. Proc. AMS 103 (2).

Engel, K., Nangel, R., 2000. One Parameter Semigroups for Linear Evolution Equations. Springer.

Gustafson, K., Rao, D., 1997. Numerical Range the Field of Values of Linear Operators and Matrices. Springer.

Haase, M., 2018. Lectures on functional analysis. In: 21st International Internet Seminar.

Haase, M., 2005. Spectral Mapping Theorems for Holomorphic Functional Calculi. London Math Sci.

Hall, B., 2013. Quantum Theory for Mathematicians. Springer.

Halmos, P., 1957. Introduction to Hilbert Space and the Theory of Spectral Multiplicity. Dover.

He'lein, F., 2014. Spectral Theory. U. of Paris.
Lumer, G., Philips, R., 1961. Dissipative operators in a Banach space. Pac. J. Math. 11 (2).
Murphy, G., 2014. C* Algebras and Operator Theory. Academic Press.
Park, E., 2013. Unbounded Operators in a Hilbert Space. Texas Christian University.
Reed, M., Simon, B., 1980. Methods of Modern Mathematical Physics, vol 1. Academic Press.
Rudin, W., 1987. 0070542341 Principles of Mathematical Analysis. Mc Graw Hill.
Shapiro, J., 2004. Notes on the Numerical Range. CARMA.
Stack Exchange, 2016. Nonempty intersection between approximate point spectrum and residual spectrum, Disintegrating By Parts.
StackExchange, 2015. Mathematics, Summary: Spectrum vs Numerical Range.
Teschl, G., 2009. 978-0-8218-4660-5 Mathematical Methods in Quantum Mechanics. AMS.
Wells, J., Williams, L., 2012. Embeddings and Extensions in Analysis. Springer.
Woozy, S., 2017. Self Adjoint Extensions of Symmetric Operators. University of Konstanz.

CHAPTER

14

Canonical commutation relations

14.1 Isometries and unitary operations

Let U be an element of the C* algebra A = B(H). U is said to be an isometry when it preserves the norm, that is, $\|U\ v\| = \|v\|$, for all v in H. In this case, it will be shown that U* U = I. In addition, when U is onto, it becomes a unitary operator, and then it is also invertible, that is, U has an inverse. However, an isometry need not be unitary. This is illustrated in Example 14.1. Also, it is an example involving generalized displacement or shift operators and hypergroups. A hypergroup is a hyperstructure involving a multivalued operation. In the hypergroup, there is a binary-type operation taking pairs of elements from a nonempty set H, into a subset of H, which is not the empty set. This operation is sometimes called addition. Very often, the set H has additional structure, such as C* algebra or Hilbert space, and so on. The next section introduces the canonical hypergroup structure using the MSA. Applications in quantum will be illustrated in a later chapter (Corsini and Leoreanu, 2003).

Example 14.1:

Let U operate on the carrier set $H=l^2$, that is, the Hilbert space of all sequences of complex numbers such that the sum of the entries' absolute values squared converges. So if $U: l^2 \rightarrow l^2$ where the operator U shifts entries to the right: U: (a0, a1, a2,...) = (0, a0, a1,...), then U is an isometry, because the norms are preserved: $\|(a0, a1, a2,...)\| = \|(0, a0, a1,...)\|$. But, U is not a unitary operator. Note that (1, 0, 0,...) is not in the range of U. So, if an inverse was to be defined for U, it could only be defined on its range, and not all of H. U is not onto, and therefore, U is not a unitary operation.#

Note that whenever U is an operator on a Hilbert space H and it is an isometry, then it is true that for any v and w in H if $\|U\ v\| = \|v\|$ then $<U\ v, U\ w> = <v, w>$. As a consequence, an isometry does preserve the inner product as well as the norm. This will be seen by showing the real and imaginary parts of two inner products are equal. First consider, $\|U\ (v+w)\|^2 = \|v\|^2 + \|w\|^2 + 2\ \text{Re}\ (<U\ v, U\ w>) = \|(v+w)\|^2$, and by substitution $\|U\ (v+w)\|^2 = \|v\|^2 + \|w\|^2 + 2\ \text{Re}\ (<v, w>)$. Therefore, $\text{Re}\ (<U\ v, U\ w>) = \text{Re}\ (<v, w>)$. Again, forming the norm squared, but this time using $\|U\ (v+i\ w)\|^2 = \|v\|^2 + \|w\|^2 + 2\ \text{im}\ (<U\ v, U\ w>) = \|(v+i\ w)\|^2 = \|v\|^2 + \|w\|^2 + 2\ \text{im}\ (<v, w>)$. In this case, it follows that im

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00028-X

(<U v, U w>) = im (<v, w>). Since the real parts as well as the imaginary parts are equal, this implies that <U v, U w> = <v, w>.

Let U be an operator on a Hilbert space H, and assume it is an isometry. That is, for any v and w in H, it follows that <U v, U w> = <v, w>. Using the definition of adjoint, namely <U v, U w> = <U* U v, w> accordingly, U*U = I, and therefore, U*U acts like the identity in a C* algebra. Notice that this is a one-sided identity. Again, let U be an isometry in C* acting on the elements in the Hilbert space H, with v and w in H. It is always 1−1, that is, U is injective. This follows because $\|U\ v - U\ w\| = \|v - w\|$, so then U v = U w iff v = w. When U is also onto, that is, surjective, then it is invertible, and therefore, UU* also equals the identity I. In this case, now, U is a unitary operator, U*U = UU* = I. Accordingly, U is a normal operator also; in other words, U commutes with its adjoint, U*. Furthermore, for nonzero v in H, $\|U^*U\ v\| = \|v\|$is greater than or equal to zero and so U*U is a positive operator. A positive operator has a spectrum always on the nonnegative part of the real line. A unitary operator always has its spectrum on the unit circle in C. Consequently, the spectrum of UU* is just the point one.

Example 14.2:

For T: H→H, where H is as usual a complex separable Hilbert space and T a positive operator, that is, <v, Tv> is greater than or equal to zero for all v in H. It follows that T is also self-adjoint. Here, it must be shown that T = T*. Since T is positive, then this implies that <v, Tv> is real valued, and so <v, Tv> = <v, Tv>* = <T v, v> = <T*v, v>. Now, let S = T − T*, as a consequence <Sv, v> = <(T−T*) v, v> = 0, but this does not show T = T*, because the inner product in this expression involves a single vector v. To conclude the verification that T = T*, the polarization identity is needed, with v and w. Set <Sv, w> = 1/4 {<S(v+w), v+w>− <S(v−w), v−w>+i<S(v+iw), v+iw>−i<S(v−iw), v−iw>} = 0. This holds for all v and w in H, thereby showing that T = T*.#

Example 14.3:

As in the previous example, let T be a positive operator, that is, <v, Tv> is greater than or equal to zero for all v in the Hilbert space H. This time, however, assume that H is a real Hilbert space, equal to R^2. So the carrier set for the Hilbert space is now a real vector space. It follows that T is not symmetric. To see this, let T =

$$\begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix}.$$

Then, if for any vector v in H v = (a b)′, then <v, Tv> = <(a b)′, (−b a)′> = 0, but T does not equal T*.#

Example 14.4:

The Cayley transform can be used in relating self-adjoint operators with unitary operators. In particular, for a self-adjoint operator T on Hilbert space H, the Cayley transform (z−i)/(z+i) is a special type of Möbius transform from complex variables. This operation

maps the x-axis onto the unit circle. It is analytic except for a simple pole at $-i$. Specifically, use the Mobius transform as a one-parameter group: $f(z) = (z+it)/(z-it)$, where t is real. So, $[(T+it)/(T-it)]^* = (T-it)/(T+it) = [(T+it)/(T-it)]^{-1}$, thus showing that f (T) is unitary.#

Similar to the illustration in the previous example, Section 14.5 is dedicated to relating self-adjoint operators with unitary operators. Indeed, here Stone's theorem shows the relationship between single-parameter families of unitary operators and self-adjoint operators.

Also, the von Neumann algebra was described in Chapter 9. However, it is interesting to relate isometries and unitarily transformations in this regard. Here, a von Neumann algebra V, which is a subset of B(H), is finite-dimensional iff all isometries are unitary transformations (Sakai, 1971).

14.2 Canonical hypergroups—a multisorted algebra view

A hypergroup is a multivalued operation with a binary-type operation taking pairs of elements from a set H into a subset of H. So it maps a pair of points into a set. It is a set-valued function. From an MSA perspective of a canonical hypergroup, there exist two sorts, and they are ELEMENTS and SUB. The first sort denotes elements from a nonempty set H; this set need not be a Hilbert space. The second sort SUB indicates all nonempty subsets from H. As in a group, there exist three signature sets one of each arity 0, 1, and 2, containing a single operation name: ZERO, MINUS, and ADD. The second sort SUB is the nonempty power set for H. Since SUB does not contain the empty set, it is denoted by $2^{H'}$. The prime indicates that $2^{H'} = 2^H - \phi$. Therefore, replace the following operator names by corresponding symbols:

ZERO, 0 is in H.
MINUS: $H \rightarrow H$.
ADD: $H \times H \rightarrow 2^{H'}$. So for a and b in H, ADD(a,b) = S where {a+b} is a subset of S.

A polyadic graph provides an illustration of these closure operations for a canonical hypergroup. See Fig. 14.1. The arity sequence is again (1, 1, 1) as in a group. However, the equational identities make a world of difference from those of a conventional group.

The corresponding operators are extended to allow for set operations. So let ZERO be represented by 0, MINUS be symbolized by $-$, and for ADD, use $+$. Also let A and B be

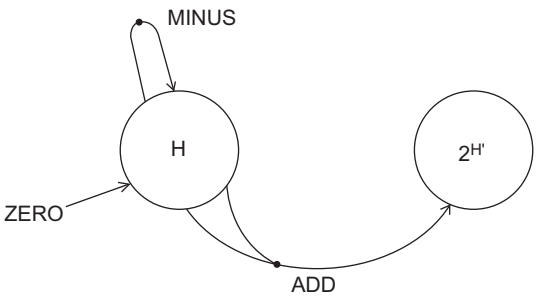


FIGURE 14.1 Canonical hypergroup polyadic graph.

nonempty subsets of H; then A+B = {a+b, for all a in A and b in B}. Note the ordering; in general, the hypergroup may not be commutative. To emphasize this, let x+A = {x+a, where a is in A} and A+x = {a+x, where a is in A}, so A+x need not equal x+A.

For the equational identities to prevail, let A, B, C, and X be nonempty subsets of H, and let x, y, and z be elements in H, then:

1) Associative: A+(B+C) = (A+B)+C. This should be understood that for all a in A, for all b in B, and finally for all c in C, the following holds: a+(b+c) = (a+b)+c.
2) Zero property: 0+x = x+0 = {x}. The meaning of this is when the binary operator is applied with x and one of its arguments is zero, then the singleton-type subset occurs, that is, then {x+0} = {x}.
3) Minus property: x+(−x) = (−x)+x = X, where 0 is an element of X. This shows that the subset X of H will contain the ZERO in H whenever either operand of ADD is the negative of the other operand.
4) Cyclic property: For x in {y+z}, this implies that z is in {x−y}. Note that liberties are taken here by using minus as a binary operation. Indeed {x−y} = {x+MINUS(y)}.

If in addition the equational identity (5) holds, then the hypergroup is said to be abelian.

5) Commutative: A+B = B+A. This should be understood that for all a in A and for all b in B, the following holds: a+b = b+a.

Example 14.5:

Consider the set H = {0, 1, 2} mod 3. Using the usual mod 3 additive structure for this carrier set, here, −2 = 1 in H and −1 = 2 in H. So, 2+1 = {0}. Furthermore, illustrating the cyclic property, 2 is in the set {1+1}; as a consequence, 1 is in the set {2−1}. This shows that hypergroup results from the module three addition operation using H and $2^{H'}$ = {{0}, {1}, {2}, {0, 1}, {0, 2}, {1, 2}, {0, 1, 2}}.#

The next example of a hypergroup is from Linzi and Stojalowska (2020).

Example 14.6:

Let H = {0, −1, 1}, and consider the addition given below; this defines what is called the sign hypergroup.

$$(-1)+(-1)=(-1)+0=0+(-1)=\{-1\}$$

$$0+0=\{0\}$$

$$1+1=1+0=0+1=\{1\}$$

$$1+(-1)=(-1)+1=\mathrm{H}.$$

Note that MINUS(1) = −1, MINUS(−1) = 1, and MINUS(0) = 0. Also, 2) holds because ZERO added to anything is anything. Additionally, 3) holds because say 1+(−1) = {0, −1, 1}, and 0 is in this set. Unlike a regular group, the ZERO element must be in a set where 1+(−1) is located, but it need not be {0}. Finally, note that 4) holds; for instance, x = 1 is in the set {1+1}, referring to 4), y = 1 and z = 1; therefore, z is in {x−y} = {1−MINUS(1)} = {1+1} = {1}.#

14.3 Partial isometries

A partial isometry U for a bounded operator on H is where U is an isometry when U is restricted to perp(ker(U)). That is, U: perp(ker(U))→ range of U. To obtain some intuition about a partial isometry operator U, it is best to see that it is a mapping from a closed subspace, A in H onto another closed subspace B of H. Often, in this case, A is referred to as the initial space, and B is called the final space. A is found by taking the orthogonal complement to the kernel, and B is found by taking the range. See Fig. 14.2. Simultaneously, the adjoint U* performs the opposite mapping. It maps subspace B onto subspace A. Additionally, it is useful to think that in the complement of domains A and B, the values are all equal to zero. U*U is a projection on subspace A, also called the initial projection, while UU* is a projection on subspace B and it is called the range projection. These mappings are used to indicate if there exist self-adjoint extensions for symmetric operators. Additionally, if there are extensions, these mappings are useful in determining these extensions.

Example 14.7:

Let L be the left shift operator in l^p, where L(v1, v2,. . .) = (v2, v3,. . .), then L is a partial isometry. Since ker(L) = {(v1, 0, 0, 0,. . .)}, it consists of all those vectors in l^p which map to zero. Perp(ker(L)) = {(v2, v3,. . .)} = ran(L).#

Example 14.8:

Let the carrier set for the sort VECTOR be the two-dimensional complex-valued field. So, here in Hilbert space H=C^2, consider the partial isometry Ua =

$$\begin{matrix} |0 & 1| \\ |0 & 0| \end{matrix}$$

Then, the initial space for Ua is A = direct sum of C and {0}. This follows because the ker(Ua) = the set of all column vectors (z 0)′, where z is any complex number. Multiplying Ua times this vector gives zero. So the initial space consists of all two-by-one column vectors with first tuple in C and the second tuple zero. Perp(ker(Ua)) = the set of all vectors of the form (0 w)′, where w is any complex number. This is also the range of Ua. The final space B is the direct sum of {0} and C.#

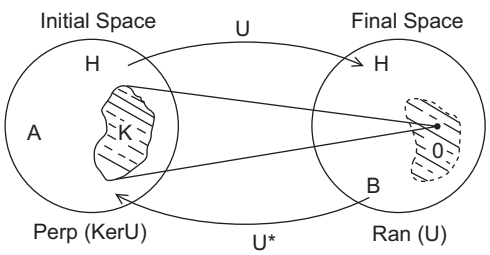


FIGURE 14.2 Partial isometry mappings.

Refer to the partial isometries from subspaces A to B in H. Denote Ua to be the partial isometry from A to B. The deficiency indices of Ua are given by the dimensions of perp(A) and perp(B), respectively. Only when these indices are equal can Ua be extended to a unitary operator on all H. First, when these indices are equal, it will be shown that U is the extension to all of H. U will equal the direct sum of Ua and Up where Up is the unitary operator from perp(A) to perp(B). This operator maps H into itself using the domain equaling the direct sum of A with perp(A) and range being the direct sum of B with perp (B). Since two Hilbert spaces are isomorphic when and only when they have the same dimension, this shows that U is an extension. On the other hand, if U extends Ua, then U (perp(A)) = perp(U(A)) = perp(Ua(A)) = perp(B); since U is unitary, dim(perp(A)) = dim (perp(B)) and the indices are equal.

The Cayley transform mentioned in Example 14.4 provides a partial isometry between ran(T + iI) and ran(T − iI) in H when T is a closed and symmetric operator. Moreover, under these conditions, the transform is unitary on H if and only if T is self-adjoint. A theorem of von Neumann states that for T closed and symmetric, T has a self-adjoint extension if and only if the deficiency indices of its Cayley transform are equal. Additionally, T has a self-adjoint extension if and only if ker(T* − iI) and ker(T* + iI), or ran(T+iI) and ran (T−iI), have the same dimension (Conway, 1999).

There are strong relations between partial isometries and projections in Hilbert space. A bounded operator U is said to be a partial isometry whenever UU* is a bounded projection. Additionally, it is called a partial isometry whenever U = UU*U or whenever U* is a partial isometry and all are in B(H). The first identity mentioned earlier is related to the pseudoinverse operation. These relations are verified below in the next section. Additionally, the pseudoinverse operation is illustrated in Section 18.4 in a machine learning environment, namely a regression application. Previously mentioned, but not in the same manner, using function composition, if Ua: A→B, then Ua*: B→A and then Ua*Ua = I on A, that is, it is the identity operator on the Hilbert space A. Similarly, UaUa* = I on B, the identity operator on B. Partial isometries are intrinsically related to generating a von Neumann algebra as well as showing the unitary equivalence of bounded operators in a Hilbert space (Pearcy, 1964).

14.4 Multisorted algebra for partial isometries

Partial isometries of B(H) operators in a C* algebra as mentioned earlier are described in several equivalent ways:

1) Idempotent: if v = UU*, then $v^2 = v$.
2) Pseudoinverse: UU*U = U.
3) Pseudoinverse: U*UU* = U*.
4) Idempotent: if w = U*U, then $w^2 = w$.

To show (1)→(2), set u = U−UU*U and then evaluate uu* = (U−UU*U) (U*−U*UU*) = UU*−UU* UU*−UU*UU*+UU*UU*UU* = v−v^2−v^2+v^3 = 0. Because $v^3 = v^2$ $v = v^2$, and

$uu^* = \|u\|^2$. Next, to show (2)→(3), take adjoint. To show (3)→(4) since $U^*UU^* = U^*$ and $w = U^*U$, multiplying the equation $U^*UU^* = U^*$ on the right by U shows $w^2 = w$. Lastly, to show (4)→(1), set $u = U^* - U^*UU^*$; next multiplying by U on the left gives $v^2 = v$. It will be seen next that in the MSA, the high view for partial isometries is that of an idempotent semigroup.

Several times in this document, it was seen that not all operators were defined fully on their domain. For instance, special attention and notation was given for the inverse operation in a field, and the same is true for both the momentum and the distance operators on a Hilbert space. Including the inversion of the zero element in a field would have enlarged the domain to include the point at infinity. This results in a pointed set, and a new algebraic structure arises, but it is not a field. Under function composition, often a partial isometry U cannot be followed by itself. That is, U^2 is not defined unless $U^*U = UU^*$, and the same is true for U^* (Paterson, 1999). As a consequence, the binary operation of multiplication, actually function composition, is modified in the MSA to accommodate these types of partial isometries. Additionally, as in previous structures, there is an order in which these operators can perform. The arguments do not necessarily commute. For these situations, as previously mentioned, the polyadic graph has slashes indicating the order of argument utilization from designated sorts. The same is true in this application along with an additional restriction, described later; also see Fig. 14.3.

For the case at hand, when the first operand or argument for the binary operator MULT is applied, there are no restrictions on this operation. Assume that the tail of the polyadic arrow has a single dash in this case. However, the second tale of the polyadic arrow is marked with two slashes. However, there is a restriction on the operand to be utilized in the present situation. To explain this, the single sort carrier set STRING must be identified. STRING is {U, U*}! And is defined as the set of all strings containing U or U* of arbitrary finite, nonzero length, but containing no two or more adjacent identical elements within a string. So star elements with nonstar elements must alternate. They cannot be adjacent within any string. For instance, UU*UU* is allowed; it is an alternating string of length four. The back b of a string is the leftmost element; however, it is applied last in function composition. The front f of a string is the right-most element of the string. For a string of length two, it is applied first in function composition. This structure becomes a semigroup using the binary operation of concatenation of strings. However, even though in this algebra multiplication is the concatenation of the operators themselves, as mentioned before, when applied to elements within a Hilbert space, they are employed in function composition. Right to left, not left to right. This order is unlike the fundamental group described previously. See Section 12.4.

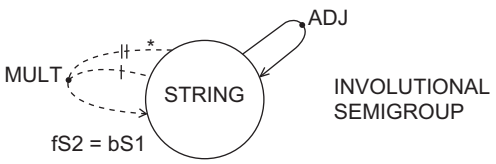


FIGURE 14.3 Polyadic graph for partial isometry [U, U*] nonzero.

Partial isometry is an idempotent using the unary operational name adjoint ADJ, along with the three equational constraints given later. First, a description of what makes a string valid.

$$\text{MULT: STRING x STRING} \rightarrow \text{STRING}$$

$$\text{ADJ: STRING} \rightarrow \text{STRING}$$

In the following denote:

Mult by $\cdot$
ADJ by *
STRING by concatenation of elements from {U, U*}! Again, this is the set consisting of one or more U, or U*, but always alternating.

The binary operation is partial because the front of the second input string must be the adjoint of the back of the first input string. Formally: f2 = (b1)*. This prevents two or more of the same elements U being adjacent to each other. The same goes for U*. For instance, say that the first argument is $\rightarrow$U*UU*UU*; then b1 is U* with an arrow pointing to it. Therefore, the second argument must have the front entry U, with no *. So say that the second entry is U*U, a valid input; then, the result of the MULT operation is U*U $\cdot$ U*UU*UU* = U*UU*UU*UU*, a string of lengthy seven. More generally, a valid string to exist will be symbolized for instance as T = u.v.w.y.z, where z is the first string. It becomes valid because the front entry in y equals the adjoint of the last entry in z. This is denoted by fy = bz*. Thus, because the individual strings y and z are multiplied to obtain a new string yz, T = u $\cdot$ v $\cdot$ w $\cdot$ yz, and then the multiplication ensures that the front of w has the adjoint value of the back of the string yz. Symbolically, fw = b(yz)* = b(y)*z, resulting in T = u $\cdot$ v $\cdot$ wyz; again, fv = b(wyz)* = b(w)*yz and so T = u $\cdot$ vwyz; finally, fu = b(vwyz)* = b(v)*wyz gives T = uvwyz.

For the polyadic graph, the second input is not only double slashed but also has the adjoint symbol * next to it. This is a reminder that the second argument must be a string whose first symbol is the adjoint of the last symbol of the first argument. As another reminder, the first argument is to the right of the second argument as in function composition, and not concatenation. Fig. 14.3 illustrates many of these facts. The equational constraints for the idempotent semigroup are as follows:

1) Idempotent Adjoint: (U*)* = U.
2) Idempotent: $(UU^*)^2 = UU^*.UU^* = UU^*$.
3) Associative: (x. (y. z)) = ((x. y). z); this is justified next.

Using formal but suggestive notation: Begin with the left hand side of 3), (x $\cdot$ (y $\cdot$ z)). As before, let f represent the front of second input, and b represent the back of the first input string. Then, fy = (bz)*, fx = (byz)*, which is exactly the same as fx = b(y)*z. This follows because the back of yz and the back of y are the same. Accordingly, (x $\cdot$ (y $\cdot$ z)) = xyz. Going the other way ((x $\cdot$ y) $\cdot$ z), since fx = (by)*, now f(xy) = xfy = (bz)* gives ((x $\cdot$ y) $\cdot$ z) = xyz again. Thus, the associative law holds; as such, this structure is more than just a groupoid; it is an idempotent semigroup.

A power partial isometry on a Hilbert space is U such that for all positive integers U^n it is also a partial isometry. All isometries along with their adjoints are power isometries. An

important example of a nonisometric power partial isometry is the truncated shift operation in the next example.

Example 14.9:

The truncated shift Tn:$C^n \to C^n$, where Tn(ej) = e(j + 1) when j <n and is zero otherwise. Here, {e1, e2,. . ., en} is a basis set for C^n. So, for instance, if n = 3, then T3(e1) = e2. Also, $(T3)^2(e1) = e3, (T3)^3(e1) = 0$. In all cases, perp(ker(T)) = ran(T).#

In general, compositions of partial isometries are rarely themselves partial isometries. It was shown that if S and T are partial isometries then so is ST, when and only when SS* and TT* commute. Moreover, even for power partial isometries, they are unitarily equivalent to the direct sum of unitary operator. These are copies of a unilateral shift S, on l^2 as well as its adjoint. Additionally, there are also copies of the truncated shift Tn given in the previous example, but for all positive integers n (Hoover and Lambert, 1974).

14.5 Stone's theorem

In short, Stone's theorem establishes a one-to-one correspondence between self-adjoint operators in a Hilbert space H and one-parameter family of unitary operators. A one-parameter family of unitary operators form a group. These operators U depend on a real parameter and are such that for 0, s, and t in R.

1) Identity U (0) = I
2) Transition U (s+t) = U (s)+U (t).

The one-parameter group in this theorem is strongly continuous in H. This is the SOT, meaning that the limit, as s→t, of $\|U(s)v - U(t)v\| = 0$, for all vectors v in H and t in R. Since the Hilbert spaces in this document are separable, unless otherwise specified, weak continuous suffices. That is WOT and the limit as s→t of <(U (s)−U (t))v, v> = 0, for all vectors v in H and t in R. In Section 9.1, descriptions of SOT and WOT are provided with examples. Nonseparable Hilbert space will be considered in a subsequent chapter. See Section 9.1, which involves operator topology. In the referenced section examples, as well as counterexamples for SOT and WOT are provided.

Of most interest is that Stone's theorem establishes that a nonbounded self-adjoint operator T with a dense domain in Hilbert space H generates a one-parameter continuous group of unitary operators: $U(t) = e^{itT}$. Moreover, all these unitary groups are generated in such a fashion. That is, there exists a densely defined self-adjoint operator T on H, such that $U(t) = e^{itT}$. For t′ in R and v in H, the limit t→t′ of U(t)v = U(t′)v. Also U(t + t′) = U(t)U(t′).

Additionally, T will be bounded iff the one-parameter group in this theorem is norm continuous in H (Hall, 2013). The Cayley transform will be employed in relating the operator T, on a dense subset dom(T), of a Hilbert space H, with the operator U given by U = (T−iI)/(T+iI). The following hold:

1) The operator U is closed iff the operator T is closed.
2) The operator U is unitary iff the operator T is self-adjoint.

Example 14.10:

Angular momentum can be considered a generator for rotation. It is a one-parameter group of unitary transformations, and by Stone's theorem, it is an infinitesimal generator performing a natural action of the rotational group SO(3,R) on $X = L^2(R^3)$. A finite rotation about an axis can be obtained by applying successive infinitesimal rotations about that axis. For instance, in R^3, a rotation about the z-axis of angle ϕ, R (ϕ), can be found using R (ϕ) = limit as $n \to \infty$ of $\left[1 - i(J\,\phi)/(n\,h)\right]^n$. Taking a logarithm, this limit can be found using basic calculus. The limit is $n \to \infty$ of $n \cdot \ln [1 - i(J\,\phi)/(n\,h)]$ equals limit as $n \to \infty$, $\ln [1 - i(J\,\phi)/(n\,h)]/(1/n)$ equals limit as $n \to \infty$ $[i(J\,\phi)/(n^2\,h)]/(-1/n^2) = -i(J\,\phi)/(h)$. Here, J is an infinitesimal symmetric, rotation operator. Raising e to the power involving the last limit gives $R(\phi) = e^{-iJ\phi/h}$.#

14.6 Position and momentum

The canonical commutation relations (CCR) express the relationship between the two principal observables in quantum, namely position and momentum. Momentum produces operations that directly affect position by way of translational changes. The basic theory behind the connection is that infinitesimal changes in position are generated by momentum. The Stone-von Neumann theorem expresses the relationship between commutation of position and momentum and the Weyl exponential relation are the fundamental cornerstones.

Example 14.11:

Consider the simplest Hilbert space H, where the carrier sets for both SCALAR and VECTOR are the one-dimensional real field. Let v be a simple element in H, commonly called a state, so $\|v\| = 1$. In this example, the basic relationship between position and momentum will be illustrated using the essence of the CBH construction. The average initial position of a particle is given by its expected value: $xo = <v\,|\,Y\,|\,v>$. When a translation by x occurs, correspondingly a new state w is produced generated by momentum P. Additionally, the expected value results in a new position. In this case, $w = e^{(-ixP/h)}$, and the new position is $X = <v|e^{(ixP/h)}Y\,e^{(-ixP/h)}|v>$. Next utilize power series for each of the exponentials. Follow this by substituting for the Lie bracket [P, Y] = −ih wherever possible as in CBH, see Example 11.3. This gives:

$X = <v\,|\,Y + [ixP/h, Y] + 1/2!\,[ixP/h, [ixP/h, Y]] + \ldots\,|\,v>$. All high-order terms have Lie brackets within Lie brackets just as in the 1/2! term. Since the Lie bracket always equals −ih, a constant, it follows [ixP/h, ih] = 0, so $X = <v\,|\,Y + x + 0 + \ldots\,|\,v> = xo + x$ and the particle moved from xo to xo plus x. The constant h in the aforementioned equation is the reduced Plank's constant.#

In the CCR structure, it is assumed that both the momentum operation P and multiplication or distance function Q are defined on a common dense domain D, in a separable Hilbert space H. They are both self-adjoint operators. Also, for the rest of this section, omitting the h factor will be in effect. Thus, it follows that [P, Q] = −iI, on D. The creation $a^{\dagger}$, and annihilation a, operators for use in Fock space are described in detail in the next

chapter. For the moment, these operators will now be represented utilizing P and Q. The same is true when describing the harmonic oscillator, in Section 19.1. Indeed, let the creation operator be given by $a^\dagger = (Q - iP)/2^{1/2}$, and the annihilation operator, $a = (Q + iP)/2^{1/2}$. It follows that $[a, a^\dagger] = 1$, to see this using the Lie bracket $[(Q+iP)/2^{1/2}, (Q-iP)/2^{1/2}] = (Q+iP)/2^{1/2}(Q-iP)/2^{1/2} - (Q-iP)/2^{1/2}(Q+iP)/2^{1/2} = 1/2i(P\,Q - Q\,P - Q\,P + P\,Q) = i[P, Q] = I$. Similarly, $[a, a] = [a^\dagger, a^\dagger] = 0$.

These identities are important in describing bosonic particles in the CCR C* algebra. Bosonic particles include photons, phonons, mesons gravitons, and several other particles. Similarly, fermions are described in the canonical anticommutative relations C* algebra. In this case, the Poisson-type bracket is employed in showing the relation between the creation $b^\dagger$ and the annihilation operation b. Here, $\{b, b^\dagger\} = bb^\dagger + b^\dagger b = 1$. Additionally, $\{b, b\} = \{b^\dagger, b^\dagger\} = 0$, and the operators b and $b^\dagger$ are bounded unlike a and $a^\dagger$. Fermion family include electrons, protons, neutrons, muons, neutrinos, and several other particles.

14.7 The Weyl form of the canonical commutation relations and the Heisenberg group

The Weyl form for the CCR is related to both the Stone's theorem and the CBH derivation. The Weyl structure utilizes two one-parameter unitary groups U(t) and V(s), similar to Stone's theorem. The functional calculus establishes the fact that for t and s, real-valued e^{isP} and e^{itQ} are unitary and bounded operators (Putman, 1967). Finally, from the CBH type of formal computation using $U(t) = e^{itQ}$, $V(s) = e^{isP}$ and for all real values of t and s, it follows that $U(t)V(s) = e^{-ist}V(s)U(t)$. The substance of the Stone-von Neumann results follows. It declares that all pairs of irreducible one-parameter unitary groups U(t) and V(s) satisfying the Weyl form of the CCR on a separable Hilbert space H are unitarily equivalent. In this case, there exists a unitary operator $W: L^2 \to H$, such that $W^*U(t)W = e^{itQ}$ and $W^*V(s)W = e^{isP}$.

Example 14.12:

The Heisenberg algebra h is closely related to CCR. It is the Lie algebra associated with the continuous nilpotent Heisenberg group of 3 by 3 upper triangular matrices, over the reals. This Lie group is denoted by H(3,R). It will be seen that the Stone-von Neumann theorem describes unique, up to isomorphism, irreducible representations of H(3,R). The algebra h has a basis X, Y, and Z that obey the commutator relations similar to those of position and momentum. The basis is given below. Here, X, Y, and Z are as follows:

$$\begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{vmatrix} \quad \begin{vmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}.$$

The commutation relations are as follows: $[X, Y] = X\,Y - Y\,X = Z$, $[X, Z] = 0$, and $[Y, Z] = 0$. The Lie algebra h is also nilpotent.#

The corresponding group that is most basic among all the Heisenberg groups is described next.

Example 14.13:

The Heisenberg group H(3,R) is the noncommutative group of all upper triangular matrices A, over the real field with the usual matrix operations. So IDENTITY = I. All equational identities for a group hold. Below is matrix A and its inverse INV(A), in that order:

$$\begin{matrix} |1 & a & b| \\ |0 & 1 & c| \\ |0 & 0 & 1| \end{matrix} \quad \begin{matrix} |1 & -a & ac-b| \\ |0 & 1 & -c| \\ |0 & 0 & 1| \end{matrix} .$$

Following is matrix B in H(3,R) along with the product MULT(A,B)

$$\begin{matrix} |1 & d & e| \\ |0 & 1 & f| \\ |0 & 0 & 1| \end{matrix} \quad \begin{matrix} |1 & a+d & b+e+af| \\ |0 & 1 & c+f| \\ |0 & 0 & 1| \end{matrix} .$$

More generally, there are Heisenberg matrices of dimension $2n + 1$, $n = 1, 2, 3, \ldots$. They form a simply connected Lie group of $(n + 2)$ by $(n + 2)$ matrices. A typical matrix is A:

$$\begin{matrix} |1 & v' & b| \\ |0 & In & w| \\ |0 & 0 & 1| \end{matrix} .$$

Here, v and w are n by 1 column vectors; v' is the transpose of v, a row vector, and In is a n by n identity matrix.#

14.8 Stone-von Neumann and quantum mechanics equivalence

Stone-von Neumann theorem is important in showing equivalence between the Heisenberg formulation of quantum mechanics and that of Schrodinger. In particular, the Heisenberg representation keeps state vectors independent of time changes, whereas observables and other operators are time dependent. The Schrodinger characterization is that the state vector evolves with time. Unitary operators U(t, to) produce time evolution of kets $|v(to)>$, from time to to a later time t. It occurs using a transition operation: $|v(t)> = U(t, to)|v(t)>$. The Schrodinger equation involves the Hamiltonian H and is $ih\partial|v>/\partial t$, that is, $ih\partial(|v(t)>)/\partial t = H|v(t)>$. When time independence exists, the partial derivative of H with respect to t is zero. The time evolution operator U(t) must obey $ih\partial(|U(t))/\partial t = HU(t)$. A more in-depth account of the Schrodinger representation of the CCR will be described, because P and Q are unbounded, and therefore domain issues must be stated. The representation was shown to be unique up to unitary equivalence by both Stone and von Neumann (Hall, 2013).

The domain for functions f in the Schrodinger formulation is S, a subspace of $L^2(R)$ consisting of functions of rapid decrease such as the Schwarz functions S(R). It could also be functions with a continuous nth derivative, C^n, or even absolutely continuous functions AC. For instance, for the momentum operator P, P: S → S, absolute continuity is sufficient, with the almost everywhere existing derivative also in L^2. An additional domain for P is Dp being the set of all functions p(x) in $L^2(R)$ with wF(p)(w) in $L^2(R)$ where F(p)(w) is the Fourier transform of p(x). Moreover, for the position operator Q, Q: S → S, also needed is

that the multiplication operation t f(t) is in L^2. Therefore, when working with both operators simultaneously, the intersection D of these type of domains must be employed. Moreover, assume that Q and P are self-adjoint or just essentially self-adjoint on the closure of this domain. The operators P and Q satisfy the CCR on this domain, that is, [P, Q] (f) = −if on D. Additionally, [P, P] = [Q, Q] = 0.

Example 14.14:

Consider f in domain $D=C^1$, as stated earlier. Using the annihilation operation a on f gives $a\ f=(Q+iP)f/2^{(1/2)}=(t\ f+f')/2^{(1/2)}=0$. The result is a first-order linear differential equation whose solution is $f=c\ e^{-t^2/2}$. Moreover, this solution is related to the ground state for the harmonic oscillator. See Section 6.1.#

The vacuum state $|0>$ is the state of lowest energy. It is also referred to as ground state or zero point field. Here, the annihilation operation has zero for the expected value in this state. The same is true for electrical and magnetic fields as well as vector potential; they have an expected value of zero. However, the expected value of the square of the field operators is nonzero. In field theory, the vacuum is the vector with no particles. Also in the Garding-Wightman axioms, the vacuum state is postulated as a unique, Poincare-invariant state (Wightman, 1976). Moreover, the vacuum is assumed to be a cyclic vector. The vacuum state is a coherent state; this means that it is a unique eigenstate of the annihilation operation a with corresponding eigenvalue a^, that is, $a\hat{}|a>=a|a>$. In other words, a coherent state remains unaltered by the annihilation of field effects.

14.9 Symplectic vector space—a multisorted algebra approach

A symplectic vector space is a real or complex vector space with an additional binary operation named SKEW. All the other operators and equational identities for the field and vector space hold true. The new operator is bilinear and such that:

SKEW: VECTOR x VECTOR → SCALAR.

The polyadic graph for this structure is provided in Fig. 14.4, where only arrows involving sort VECTOR are illustrated. The additional equational identities are given by first replacing:

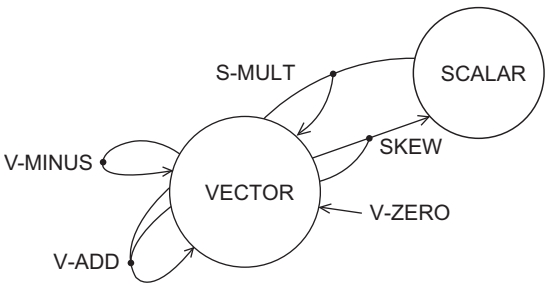


FIGURE 14.4 Polyadic graph for a symplectic vector space.

VECTOR by u, v.
SKEW by S.

The identities are as follows:

1) Alternating S (u, u) = 0.
2) Nondegenerate S (u, v) = 0, for all v implies that u = 0.

Note that S (u, v) = − S (v, u). This follows by using S (u+v, u + v) = S (u, u) + S (u, v)+S (v, u)+S (v, v) = 0, so S (u, v)+S (v, u) = 0.

Example 14.15:

An abstraction of the Heisenberg group can be formed from a symplectic vector space V. Here, for a real-valued vector space, the product space V x R is the domain. For v and w in V and t and r in R, the group law is (v, t) x (w, r) = (v+w, t+r+S (v, w)/2), where S is the operation corresponding to SKEW. In terms of the MSA characterization:

$$\text{IDENTITY} = (0, 0)$$

INVERSE (v, t) = (− v,− t), since S (v,− v) = − S (v, v) = 0, as a consequence of the bilinearity.

MULT ((v, t), (w, r)) = (v+w, t+r+S (v, w)/2), note that MULT ((v, t), (−v,− t)) = (v−w, t−t+S (v, − v)/2) = (0, 0).

As a carrier set, consider sort VECTOR to be $V = R^2$ with the usual operations corresponding to the vector space signature sets. Additionally, assume that the operation associated with SKEW is S (v, w) = v′ H w. Here, v and w are 2 by 1 real vectors, v′ is the transpose of v, and H is a skew symmetric 2 by 2 matrix with zeros on the main diagonal; thus H′ = −H.#

Example 14.16:

The abstract Heisenberg algebra can be formed from a symplectic vector space V. Here, for a real-valued vector space, the product space V x′ R is the domain, where x′ denotes the direct product. For v and w in V and t and r in R, the commutator law is [(v, t), (w, r)] = (0, S (v, w)), where S is the operation corresponding to the binary operation in the symplectic vector space SKEW.#

The strong operator topology plays an important role in the representations of a Heisenberg group; see Section 9.1. This representation is continuous from the Heisenberg group to a topological group of unitary operators on a Hilbert space H. The Schrodinger representations form a family of irreducible infinite dimensional representations of the Heisenberg group. For a Heisenberg algebra, a representation is a Lie algebra homomorphism involving skew-symmetric endomorphisms for a dense subspace of a Hilbert space H. The representation is irreducible whenever H is not {0} and all invariant closed subspaces of H are H or {0} subspace. All Schrodinger representations of both the Heisenberg group and algebra are irreducible (Hall, 2013).

14.10 The Weyl canonical commutation relations C* algebra

Weyl operations determine the C* algebra called the Weyl CCR C* algebra; it involves a real symplectic Hilbert space H, generated by elements {W(f) | f is in H} and with the equational identities holding:

1) W(0) = 1.
2) W(f) unitary.
3) W*(f) = W(− f), f in H.
4) $W(f)W(g) = e^{(-iIm(<f,g>)/2)}W(f+g)$, g also in H. Note that S(f, g) = (− i Im(<f, g>)/2) is a symplectic bilinear form. For instance, since $<f, f> = \|f\|^2$ is real, the imaginary part is equal to zero. Also, when <f, g> = 0 for all g in H, then f also must equal zero.

The Weyl version of the CCR C* algebra is unique up to * isomorphism (Slawny, 1972). Moreover, it was shown that for f not equal to g, $\|W(f)-W(g)\|$ is greater than or equal to $2^{1/2}$. So the unitary group here cannot be norm continuous, and the C* algebra is not separable. This C* algebra is faithfully represented on the symmetric Fock space. This space has a distinguished vector called the vacuum vector. As mentioned previously, it is destroyed by the annihilator operation and is cyclic with respect to the creation operator. Recall the cyclic property in this context means that successive applications of the creation operator generate all states. Bosonic Fock space is described in the next chapter.

Weyl rewrote the CCR in terms of unitary operations and unitary groups. Recall the Schrodinger representation. For the operators of position Q(f) = tf, and for momentum P(f) = −if′, they are both defined as essentially self-adjoint on the closure of domain D in H. The corresponding bounded unitary operators are $U(t)=e^{itP}$ and $V(s)=e^{iQs}$. The critical relations involving the bounded unitary operators are (U(t)f)(x)= f(x+t), and $(V(s)f)(x)=e^{isx}f(x)$. These identities can be formally verified by employing Taylor series evaluated at the origin. Doing this, and using O to denote the remainder of terms that are assumed to be small, gives the following: First, (U(t)f)(x) = [1 + itP + O]f(x) = [f(x) + tf′(x) + O] = f(x + t) = [f(x) + f′(x)t/1! + O]; Next, (V(s)f)(x) = [1 + iQs + O]f(x) = [f(x) + isxf(x)] = e^{isx}f(x) = [1 + isx + O]f(x).

Apply the two identities found earlier. Using V on the first approximation shows that $(V(s)U(t)f)(x)=(V(s)f)(x+t)=e^{is(x+t)}f(x+t)$. Next applying U to the second approximation mentioned earlier gives $(U(t)V(s)f)(x)=e^{isx}(U(t)f)(x)=e^{isx}f(x+t)$. Comparing these two results yields the desired conclusion: $e^{ist}V(s)U(t) = U(t)V(s)$, the Weyl relation for the CCR. This formal development is attributed to Schrodinger.

The Weyl relation is a strongly continuous unitary group on a Hilbert space; the representation is irreducible and unique up to unitary equivalence. This means there is no sub-Hilbert space K, in H with the following properties: Here, K is nonzero, not equal to H; U(t)K is a subspace of H; and V(s)K is a subspace of H for all real s and t. When there is a change of basis, a new representation forms (U′, V′) on H′. But since there is unitary equivalence, there is a unitary operator W, W: H→H′ such that U(t) = W* U′ (t) W and V(s) = W* V′ (s) W. All Weyl relation representations of the CCR are unitarily equivalent to an at most countable direct sum of Schrodinger representations. An irreducible representation of the Weyl relation is unitarily equivalent to the Schrodinger representation (Slawny, 1972).

The Weyl CCR C* algebra is often denoted by CCR(H) or CCR(H, s) where H is for Hilbert space and s stands for symplectic space, Im(<f, g>). CCR(H) is a simple C* algebra, that is, it does not have any two-sided ideals. Additionally, let T be an isometry such that T: H→H, and assume that it preserves symplectic structure. That is, s(T f, Tg) = s(f, g), for f and g in H. Under these conditions, T can be implemented as a star automorphism A, from CCR(H) to CCR(H). Additionally, there is a unique automorphism B, from CCR(H) to CCR(H) such that B(W(f)) = B(W(T f)).

References

Conway, J., 1999. 0-8218-2065-6 A Course in Operator Theory. AMS Bookstore.
Corsini, P., Leoreanu, V., 2003. Applications of Hyper Structure Theory. Kluwer, Dordrecht.
Hall, B., 2013. 978-146471158 Quantum Theory for Mathematicians, vol 267. Springer.
Hoover, T., Lambert, A., 1974. Equivalence of power partial isometries. Duke Math. J. 41 (4).
Linzi, A., Stojalowska, H., 2020. Hyper valuations on hyper fields and ordered canonical hypergroups. arXiv:2009.08954v1.
Paterson, A., 1999. 0-8176-4051-7 Groupies, Inverse Semigroups, and Their Operator Algebras. Birkhäuser.
Pearcy, C., 1964. On certain von Neumann algebras which are generated by partial isometries. JSTOR 15 (3), AMS.
Putman, C., 1967. Commutation Properties of Hilbert Space Operators. Springer-Verlag.
Sakai, S., 1971. 3-540-63633-1 C* Algebras and W* Algebras. Springer.
Slawny, J., 1972. Algebra of Canonical Commutation Relations. Project Euclid.
Wightman, A., 1976. Hilberts Sixth Problem: Mathematical Treatment of the Axioms of Physics, Vol 28. AMS.

CHAPTER

15

Fock space

15.1 Particles within Fock spaces and Fock space structure

Fock space is a direct sum of tensor products of Hilbert spaces. It is used in representing various states of identical particle Hilbert spaces H. All elements are indistinguishable from one another in a fixed Fock space. The major types of particles described in Fock spaces are bosons and fermions. Other particles such as negatively charged ions, like anyonic particles, are not described in this space. For bosons, a sum of a symmetrized tensor product of n single-particle Hilbert spaces is employed. Zero-particle states, single-particle states, two-particle states, ... These are all a tensor product of H with itself. All these products exist in Fock space. Bosonic Fock space is a representation of a Weyl algebra. For fermions, an anti-symmetrized tensor product of single-particle Hilbert spaces is used. Fermionic Fock spaces provide a representation for a Clifford algebra. In both cases, a Hilbert space completion is utilized in forming the direct sum of these tensor products. The inner product of elements in Fock space is defined as sums and products of inner products of individual Hilbert spaces.

Example 15.1:

For vectors v1 and w1 both in Hilbert space H1 and vectors v2 and w2 in Hilbert space H2, the inner product of vectors in the tensor product H1⊗H2 is given by $<$v1⊗v2, w1⊗w2$>$ = $<$v1, w1$>$ $<$v2, w2$>$. This is the inner product from the individual Hilbert spaces multiplied together as seen in Section 7.4.#

In general, the form of Fock space is F = the direct sum, $\sum_{n=0}^{\infty}\left[S(\otimes H)^n\right]$. Here, S is a function performing symmetric or antisymmetric operations on a tensor depending on whether they are bosons or fermions, respectively. The quantity $(\otimes H)^n$ is the tensor product of H, n times. In short, Fock space F will be written in this section in the following manner: It is F = (+) [S $(\otimes H)^n$]. Here, n = 0, 1, 2,..., and the operator (+) stands for the direct sum from 0 to infinity. So, expanding the above expression in more detail gives F = C+H+S (H⊗H)+S (H⊗H⊗H)+.... In this representation, the plus, +, is used for the direct sum of two individual terms. The quantity C is the field of complex numbers denoting the vacuum state, and it is abbreviated vac. Vac is the state consisting of no particles. The Hilbert space H is the state for single particles, while S (H⊗H) describes the states for

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00029-1

two identical particles. The two identical particle state is also denoted by S H2. Similarly, S (H⊗H⊗H) describes the states of three identical particles, and it will be denoted by S H3, and so on.

The Hilbert space completion of the algebraic direct sum above is required. In this direction, (+) Hn = {(x0, x1, x2, ...)}, where xj is in Hj. Additionally, the sum x0+x1+x2+... must be such that the sum $\sum_{n=0}^{\infty} \|xn\|^2 =$ the sum $\sum_{n=0}^{\infty} \langle xn, xn \rangle$, and this sum is less than infinity. Finally, identify a vector v in Hj by the jth tuple of (0, 0, ..., 0, v, 0, ...), and use the notation v = vj. Also, the space Hj is orthogonal to the space Hi whenever j does not equal the integer value i. The latter statement means that the n particle subspaces for different values of n are orthogonal.

It is useful to provide an additional, but related concept of a state for Fock space. Begin with the symmetric Fock space; it must always be infinite dimensional. For a finite-dimensional Hilbert space H1, consider an observable having eigenvectors |ej>, j = 1, 2, ..., n, and corresponding eigenvalues aj. When a single particle occupies the kth state, this means that the eigenstate ket, |ek> = |0, 0, ..., 0, 1, 0, ... 0>, and where the one appears in the kth position. Correspondingly, with similar assumptions, the two-particle space H2 could have both particles in a single state, or a single particle in two distinct states among the n states. So, for instance, in bosonic Fock space, the possibilities for two particles in a single state are both in the first state, |2, 0, ..., 0>, or both in the second state, |0, 2, 0, ..., 0>, and so on to the last state |0, ..., 0, 2>. Additionally, in any Fock space, the two particles can be separated: |1, 1, 0 ..., 0>, |1, 0, 1, 0, ..., 0>, ..., |0, ... 0, 1, 1>. These are all the possible state occupancies for two particles. An important thing to realize is that H2 is not the tensor product of H1 and H1. It is isomorphic to the symmetrized direct sum of tensor products, that is, H = S (H1⊗H1). Briefly, in this case, S (ha⊗hb) = (ha⊗hb)+(hb⊗ha). Similarly, in H3 there would be a direct sum of three factorial tensored products, that is, six possibilities. Linear combinations of product states are the general state in Fock space.

Example 15.2:

Consider H2; this is a two-particle Hilbert space, using distinct kets {|0>, |0′>}. As seen previously, there exist four possible states involving these two kets; these distinct states are given by {|0 0>, |0 0′>, |0′ 0′>, |0′ 0>}. Each state has an equal probability of being assembled, but not observed. There exist three symmetrical states for observation and a single antisymmetrical state. These are in order, |0 0>, |0′ 0′>, |0 0′>+|0′ 0>, and |0 0′>−|0′ 0>. The first three are bosonic states; they are similar to the form, S (ha⊗hb) = (ha⊗hb)+(hb⊗ha). The last expression will be shown to be a fermionic state.#

15.2 The bosonic occupation numbers and the ladder operators

In Fock space, there exist transitions between particles within subspaces of distinct numbers. Thus, particle numbers are not preserved; particles can be created or destroyed, that is, annihilated. These elements can be added or removed from any given energy state. Operators corresponding to these functions are called ladder operations. Instances of these operators are seen in Section 14.8. In quantum mechanics, these transitions caused by the

ladder operators are referred to as second quantization and involve the blending of Hilbert subspaces Hn and H(n + −1). The ladder operators are used in several areas of physics in raising and lowering energy eigenfunctions, as well as changing units of the angular momentum number. The operators have arguments that are particles like photons or quasiparticles like phonons.

The bosonic Fock space algebra over the Hilbert space H allows all states to be occupied by n, where n is any finite number of particles. That is, n = 0, 1, …. In this space, the number operator Nj provides the count of the number of particles for the jth eigenstate. This operator keeps track of the quantity of particles that are located in every state. The occupational number nj, for an observable A, along with nonnegative integer-valued eigenvalues, is such that Nj (|n0, n1, n2,…, nj,…, nk>) = nj |n0, n1,…, nj,…, nk>. Accordingly, for the operator, Nj, the eigenvector is |n0, n1, n2,…, nj,…, nk>, and the corresponding eigenvalue is the occupational number, nj. The observable A can be written as a sum, $\sum_j$(cj Nj), where the cj are scalars.

The creation operation is employed in Fock space to add a particle in the prescribed quantum state, j. This operator is denoted by $a_j^\dagger$; it focuses on the jth state only and increments the occupancy number of this state by one. This operation on the quantum state is an exterior or symmetric multiplication. Specifically, $a_j^\dagger$ (|…, nj, …>) = $(nj+1)^{1/2}$ (|…, nj+1, …>). The parenthesis will often be left out for these operators in the following. Opposite to the creation operator is the annihilator operator aj. Examples of both of these operators were used in the CCR. See Section 14.8. There, the creation operator was given by $a^\dagger = (Q - iP)/2^{1/2}$, involving the position operator Q and the momentum operator P. Similarly, the annihilation operator a was given by $a = (Q+iP)/2^{1/2}$.

In general, the creation operator adds a particle whenever a particle is present or not. For a nonvacant state, the annihilated will decrement the occupation number by one. However, when the jth state is empty, the annihilation operator performed on this state will result in zero. When the annihilator acts on the vacuum state C, that is, vac, the result will also be zero. The operation performed by the annihilator is an interior product-type operation with the state. In summary, for nonvacant states, aj |…, nj, …> = $nj^{1/2}$ |…, nj −1,…,>, when nj is positive, and this yields zero otherwise. Fig. 15.1A illustrates the ladder operators in bosonic Fock space.

Ladder operators and matrix representation of the annihilation operation a and the creation operation $a^\dagger$ are given in matrix form below in a respective order. These matrices provide the scalar multiple for an allowable decrement or an increase in particles in a designated state. On top of the matrix above, each column is the state number on which the operation is performed. The scalar multiple to be used is found by identifying the column number, starting from zero. The corresponding row number is one less when using the annihilation operator, a. When using the creation operator, $a^\dagger$, use one more row, that is, an additional row. Remember that all states begin with state zero. Below in the following order is the matrix for the annihilation operation a, followed by the matrix for the creation operator, $a^\dagger$:

(0	1	2	3	4…)	(0	1	2	3…)
|0	1	0	0	0…|	|0	0	0	0…|
|0	0	$2^{1/2}$	0	0…|	|1	0	0	0…|
|0	0	0	$3^{1/2}$	0…|	|0	$2^{1/2}$	0	0…|
|0	0	0	0	$4^{1/2}$..|	|0	0	$3^{1/2}$	0..|

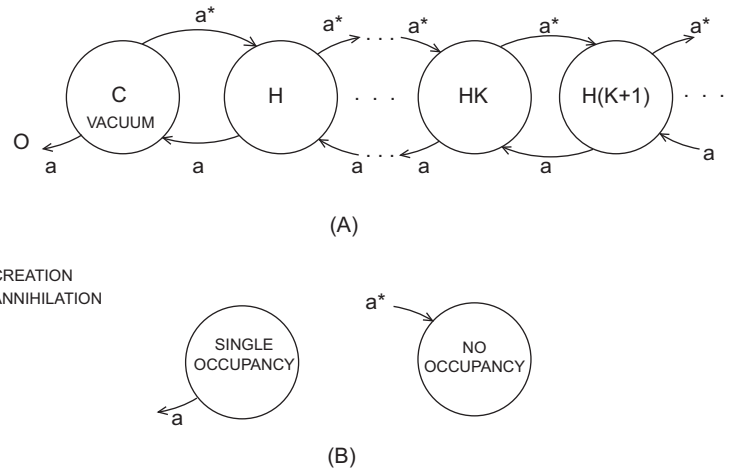


FIGURE 15.1 Ladder operators for Fock spaces. (A) Bosonic Fock space, (B) Allowable operations in Fermionic Fock space.

Example 15.3:

This is an application of using the matrices mentioned earlier to find the scalar multiple for both creating and annihilating a particle from a designated state. Say that the creation operator is $a2^{\dagger}$, that is, the state two is to be incremented by an additional particle. Go to the column two on top and to row three on the right matrix mentioned earlier, and find the scalar entry $3^{1/2}$. For instance, before applying the operation, assume that the space is occupied as $\phi = |3, 0, 6, 4, 0, \ldots, nk>$; then $a2^{\dagger}(\phi) = 3^{1/2}\ |3, 0, 7, 4, 0, \ldots, nk>$. Next, assume that an annihilation operation is to be performed on the third state involving ϕ. To find $a3(\phi)$, using the third column second row of the first matrix mentioned earlier gives $3^{1/2}$, so $a3(\phi) = 3^{1/2}\ |3, 0, 6, 3, 0, \ldots, nk>$. If $a1(\phi)$ is desired, then the answer is zero, because there is no particle to be taken away from state one in ϕ.#

The matrices mentioned earlier can be used with the number basis eigenvectors ψ_i, and ψ_j. This will be the situation in Section 19.1, where the harmonic oscillator solutions are described for all energy values. In any case, the following identities hold: $a_{i,j} = <\psi_i\ |a\ |\psi_j>$ and $a_{i,j}{}^{\dagger} = <\psi_i\ |a^{\dagger}\ |\psi_j>$.

There are numerous relationships between the creation operator, $a^{\dagger}$, the annihilation operator, a, and the number N, operator. Some of these relationships were illustrated in the previous section. Before additional relations are identified, it will be assumed that the creation operation and the annihilation operations are adjoints of one another. That is, $a^{\dagger} = a^*$ and $(a^{\dagger})^* = a$. These facts are justified in the C* algebra of the Banach space completion for the bosonic Fock space (Townsend, 2000). Now use the nonvacuum and occupied state with n particles. Here, $a^{\dagger}a|n> = n^{1/2}a^{\dagger}|n-1> = n^{1/2}n^{1/2}|n> = N|n>$. Thus, the identification between the number operator and both the creation and annihilation operations is $N = a^{\dagger}a$. In a similar fashion, $aa^{\dagger}|n> = (n+1)^{1/2}a|n+1> = (n+1)^{1/2}(n+1)^{1/2}|n> = (n+1)|n>$. It follows that $aa^{\dagger}$ acts like (N+I). This is the number operator plus the identity map. Commutation relations are a consequence of the identities

just given. The commutator, $[a, a^\dagger] = aa^\dagger - a^\dagger a = (N + I) - N = I$. Additionally, $[N, a^\dagger] = Na^\dagger - a^\dagger N = a^\dagger aa^\dagger - a^\dagger a^\dagger a = a^\dagger(aa^\dagger - a^\dagger a) = a^\dagger(I) = a^\dagger$. Similarly, $[N, a] = Na - aN = a^\dagger aa - aa^\dagger a = (a^\dagger a - aa^\dagger)a = -a$.

Example 15.4:

The ground state of zero can be algebraically explained using the ladder operators using the fact that the operators a and $a^\dagger$ are mutually adjoint. Here, for the annihilation operation energy, eigenstates can only be lowered to zero, and never less than this amount. To see that the minimum eigen number is zero, consider $N = a^\dagger a$, and $N\ \phi_n = n\ \phi_n$. Then $<\phi_n | N\ \phi_n> = <\phi_n | a^\dagger\ a\ \phi_n> = <a\ \phi_n | a\ \phi_n> = \|a | \phi_n>\|^2$, which is nonnegative and equals zero iff $a\ | \phi_n> = 0$, and so for $n = 0$, again, it is shown that $a\ | \phi_0> = 0$.

Basic Fock states with n bosons are given by the cyclic property of the creation operation: $(a^\dagger)^n|0>$, where n is a positive integer. These Fock states usually do not contain any information as to the phase. However, the bosonic coherent state v is in essence a Gaussian wave packet. It is a special bosonic state and involves the annihilation operator a. Specifically, it is the eigenstate of a. So, $a\ v = r\ v$, where r is the eigenvalue and since v is a state; it is assumed that $\|v\| = 1$. Coherent states have dynamics similar to classical oscillatory trajectories and are related to the quantum harmonic oscillations (Gazeau, 2009).

The coherent state is given by an infinite sum involving the occupation numbers. It is defined as $v = e^{-1/2|r|^2} \sum_{n=0}^{\infty} r^n/(n!^{1/2})|n>$, where $|n>$ is the number operation. To see this, let v = the sum, $\sum_{n=0}^{\infty} cn|n>$, where cn are complex scalars. Then using the annihilation operation on v, a v = the sum, $\sum_{n=1}^{\infty} cn\ a|n> =$ the sum, $\sum_{n=1}^{\infty} cn\ n^{1/2}|n\text{-}1> =$ the sum, $\sum_{n=0}^{\infty} c(n+1)(n+1)^{1/2}|n>$. Using the definition of v above, set r v equal to the sum, $\sum_{n=0}^{\infty} cn\ r|n>$. Form the inner product of both of these quantities with the bra number operator $<n\ |$. Then use the orthogonality condition $<n, n'> = 0$, except when $n = n'$; in this case, the result is one. It follows that $c(n+1)\ (n+1)^{1/2} = cn\ r$ or $c(n+1) = cn\ r/(n+1)^{1/2}$. Employing successive substitution to this difference equation yields $cn = c(n-1)\ r/n^{1/2}$ and so $c(n-1) = c(n-2)\ r/(n-1)^{1/2}$; substituting gives $cn = c(n-2)\ r^2/(n\ (n-1))^{1/2} = c(n-3)\ r^3/(n\ \ (n-1)\ \ (n-2))^{1/2} = \ldots = co\ \ r^n/n!^{1/2}$. Thus, v = the sum, $\sum_{n=0}^{\infty} cn|n> =$ the sum, $\sum_{n=0}^{\infty} \left(co\ r^n/n!^{1/2}|n>\right.$. Since $\|v\| = 1$, then $<v, v> = 1$. Forming this inner product gives v = the sum, $\sum_{n=0}^{\infty}(|co|^2|r|^{2n}/n!)$; this follows because $<n, n'> = 0$, when n, n′ differ. Therefore, $\langle v, v\rangle = |co|^2 e^{|r|^2}$. Setting this quantity equal to one gives $|co|^2 = e^{-|r|^2}$, or $co = e^{-1/2|r|^2}$. This shows that the coherent state $v = e^{-1/2|r|^2} \sum_{n=0}^{\infty} r^n/\left(n!^{1/2}\right)|n>$.

Example 15.5:

Consider two coherent states v and w, where v is given by the aforementioned expression and w is given in an identical manner as v, for instance, $w = e^{-1/2|s|^2} \sum_{n'=0}^{\infty} \left(s^{n'}/\left(n'!^{1/2}\right)\right)|n'>$. Then $\langle w, v\rangle = e^{-1/2|r|^2} e^{-1/2|s|^2} e^{rs*}$. This expression follows by substituting into the inner product for both w and v. Then, the only term needed to show is e^{rs*}, because the other terms are scalars. The last term follows by Taylor series expansions for both w and v. Then noticing that $<n', n> = 0$ except when $n' = n$, here it is one. This reduces the double sum to a single sum. In this situation, the expansion becomes the sum,

$\sum_{n=0}^{\infty} (r\, s^*)^n/n!$, which in conclusion is formally given by e^{rs^*}. Special cases arise: For $r = s$, then $<w, v> = 1$. Also, for $r = -s$, then $<-w, v> = e^{-2|r|^2}$. For larger $|r|$, v and w become close to orthogonal.

Now, the solution for defining the difference equation for the adjoint operator will be found. Recall $a^\dagger|n-1>/n^{1/2} = |n>$, and the solution is not a surprise that any occupation number can be found from the vacuum state. Thus, $|n> = a^{\dagger n}/n!^{1/2}|0>$. To see this, as before, iterate $a^\dagger|n-2>/(n-1)^{1/2} = |n-1>$; then substituting for $|n-1>$ into the equation, $a^\dagger|n-1>/n^{1/2} = |n>$, gives $a^{\dagger 2}|n-2>/(n(n-1)^{1/2})$. Continuing to iterate and substitute provides the desired results via induction.#

15.3 The fermionic Fock space and the fermionic ladder operators

The occupation number in the case of fermionic Fock space is trivial because N can only be zero or one. Either there exists a particle in a given state or there is none. The occupation operator is a binary function in the fermionic Fock space. The ladder operators $a^\dagger$ and a in the fermionic Fock space are similar to those in the bosonic Fock space. Intuitively, as before, the creation operator $a^\dagger$ adds a particle to a state, but in this space the state must have to be empty. The annihilation operator removes a particle from a state, provided there is a single particle to be annihilated. This is also illustrated in Fig. 15.1B. The anticommunication relations providing the fermionic Fock space algebra are given by the Poisson bracket: {ai, aj} = ai aj+aj ai = 0, for all states i and j, also $\{a_i^\dagger, a_j^\dagger\} = 0$, and finally, $\{a_i, a_j^\dagger\} = 0$, except that it equals one when i = j. As a consequence of the first two identities, it follows that $aa = a^\dagger a^\dagger = 0$. The number operator as in the bosonic case is given by $N = a^\dagger a$. However, since $\{aa^\dagger + a^\dagger a\} = 1$, then $aa^\dagger = 1 - N$. From this, $N(1-N) = a^\dagger aaa^\dagger = 0$. So the eigenvalues for N are either zero or one. This confirms the Pauli principle that the occupation number is zero or one. It can be seen that the eigenstates of N are $a^\dagger\ |0>$ and $a\ |1>$ with corresponding eigenvalues 1 and 0, respectively. These results follow since $(1-N)\ a^\dagger\ |0> = a\ a^\dagger\ a^\dagger\ |0> = 0$, as well as $N\ a\ |1> = a^\dagger\ a\ a\ |1> = 0$.

In two-dimensional Hilbert space with orthonormal basis, {|0>, |1>}. It follows that $a\ |0> = a^\dagger\ |1> = 0$, $a^\dagger\ |0> = |1>$; finally, $a\ |1> = |0>$. Note that $a = |0><1|$, because $a\ |0> = |0><1|0> = 0$, and also $a\ |1> = |0><1|1> = |0>$. Correspondingly, for the creation operation $a^\dagger = |1><0|$, since $a^\dagger\ |1> = |1><0|1> = 0$, and $a^\dagger\ |0> = |1><0|0> = |1>$. Also, $a\ a = |0><1|0><1| = 0$, and $a^\dagger\ a^\dagger = |1><0|1><0| = 0$. As a consequence, neither a nor $a^\dagger$ is self-adjoint; however, they are conjugate adjoint. That is, $(a)^* = a^\dagger$, because $(|0><1|)^* = |1><0|$; similarly, $(a^\dagger)^* = a$.

In the Hilbert space C^2, it was seen that the representations for |0> and |1> are given by the column vectors $(1\ 0)'$ and $(0\ 1)'$, respectively. In this space, two-by-two matrices for the creation operation $a^\dagger$ and annihilation operation a are given in order as follows:

$$\begin{matrix} |0 & 0| & |0 & 1| \\ |1 & 0| & |0 & 0|. \end{matrix}$$

Using these matrices and the column vectors above, again it is seen that $a^\dagger\ |0> = (0\ 1)' = |1>$, $a^\dagger\ |1> = 0$, $a\ |0> = 0$, and $a\ |1> = (1\ 0)' = |0>$.

In a fermionic Hilbert space with n states, there are 2^n dimensions. Moreover, each transposition of elements results in an overall negation. This follows due to the anticommutation of creation operations $a^\dagger j$ and $a^\dagger k$. That is exchanging j and k creates an opposite sign, from+to−or conversely. For instance, $|nj, nk\rangle = aj^\dagger ak^\dagger |vac\rangle = -|nk, nj\rangle$. Here vac is the vacuum state.

Example 15.6:

In H3, this is a Fermionic Fock space with three distinct allowable states. The Hilbert space consists of potentially three identical fermions. So, $|nj, nk, nm\rangle$ spans the space. In this case, the distinct possibilities are eight in number. They are $|0, 0, 0\rangle$, $|0, 0, 1\rangle$, $|0, 1, 0\rangle$, $|1, 0, 0\rangle$, $|0, 1, 1\rangle$, $|1, 1, 0\rangle$, $|1, 0, 1\rangle$, and $|1, 1, 1\rangle$. Also since $|nj, nk, nm\rangle = aj^\dagger ak^\dagger am^\dagger |vac\rangle$, then $ak^\dagger aj^\dagger am^\dagger |vac\rangle = -aj^\dagger ak^* am^\dagger |vac\rangle$. However, notice that $|ak^\dagger am^\dagger aj^\dagger |vac\rangle = aj^\dagger ak^\dagger am^\dagger |vac\rangle$, since there are two transpositions. In this case, the sign changes twice.#

For a general fermionic Fock state $|f\rangle = |n1, n2, \ldots, nj, \ldots, nk, \ldots\rangle$, the creation operation is such that $aj^\dagger |f\rangle = |n1, n2, \ldots, nj+1, \ldots\rangle$ when $nj = 0$, and the result is zero otherwise. The annihilation operation using $|f\rangle$ above is such that $aj |f\rangle = |n1, n2, \ldots, nj-1, \ldots\rangle$; this occurs only when $nj = 1$, and the result is zero otherwise. To move a particle position from the jth location to an empty location say k, then an annihilation must be performed on the particle in the jth position and a creation of a particle on the kth position. In this strict order, and do not forget, all particles here are identical. So, $ak^\dagger aj |f\rangle = |n1, n2, \ldots, nj-1, \ldots nk+1, \ldots\rangle = |g\rangle$. Note that $aj\, ak^\dagger |f\rangle = -ak^\dagger aj |f\rangle = -|g\rangle$.

Example 15.7:

In fermionic Fock space H2, the resolution of the identity is given by $I = \{a, a^\dagger\} = a\, a^\dagger + a^\dagger a = |0\rangle\langle 1|1\rangle\langle 0| + |1\rangle\langle 0|0\rangle\langle 1| = |0\rangle\langle 0| + |1\rangle\langle 1|$. Notice that $a\, a^\dagger I = a\, |1\rangle\langle 0|(|0\rangle\langle 0| + |1\rangle\langle 1|) = a\ |1\rangle\langle 0| = |0\rangle\langle 1|1\rangle\langle 0| = |0\rangle\langle 0|$; this is a projection onto the first component of I. Also, $a^\dagger\, a\, I = a^\dagger\ |0\rangle\langle 1|(|0\rangle\langle 0| + |1\rangle\langle 1|) = a^\dagger\ |0\rangle\langle 1| = |1\rangle\langle 0|0\rangle\langle 1| = |1\rangle\langle 1|$; this is a projection onto the second component of I.#

Consider C^2, but this time let $|0\rangle = (1\ 0)'$ denote the vacuum state. And let $|1\rangle = (0\ 1)'$ represent the occupied state in this Fock space. The number operator n, the creation operator $a^\dagger$, and the annihilator operation a are all illustrated as the 2 by 2 matrices provided in the following order:

$$|0\ \ 0|\quad |0\ \ 0|\quad |0\ \ 1|$$
$$|0\ \ 1|\quad |1\ \ 0|\quad |0\ \ 0|.$$

Note that $a^\dagger a = n$, by multiplying the second matrix above by the third to yield the first matrix. The Poisson bracket operator gives $\{a, a^\dagger\} = aa^\dagger + a^\dagger a = I$. As well as $a^\dagger |0\rangle = |1\rangle$, $a^\dagger\ |1\rangle\ 0$, $a\ |0\rangle = 0$, and finally $a\ |1\rangle = |0\rangle$.

15.4 The Slater determinant and the complex Clifford space

Vectors within fermionic Fock space can be represented by determinants called Slater determinants. The antisymmetry results from the definition of determinants. Beginning with two-dimensional space, a review of notation will be conducted. A two-particle state in H2 can be written involving two single-particle states, |n1> and |n2>. First, assume that particle r1 = 1 is in state |1> and particle r2 = 2 is in state |2>; then this will be written as follows: (r1|n1)(r2|n2) = f(r1)f(r2). Similarly, assume that particle r2 is in state |1> and particle r1 is in state |2>; then this will be written as follows: (r1|n2)(r2|n1) = f(r2)f(r1). Here, particles are indistinguishable, and because they are antisymmetrical, the allowed representation is $|v> = 1/2^{1/2}$ (|n1>|n2>−|n2>|n1>) = $1/2^{1/2}$ (f1(r1)f2(r2)−f2(r1)f1(r2)). The latter equation is the determinant. Specifically, the latter representation can be written as a 2 by 2 Slater matrix S 2, which is given below with the scalar $2^{1/2}$ in front, so $2^{1/2}$S2 = :

$$\begin{vmatrix} f1(r1) & f1(r2) \\ f2(r1) & f2(r2) \end{vmatrix}.$$

Generalization to N particle antisymmetric wave functions is given by the N by N Slater determinant S N (Slater, 1929). The subspace spanned by these states is the fermion Fock space. The matrix S N is given as $N!^{1/2}$ SN =

$$\begin{vmatrix} f1(r1) & \cdot & \cdot & \cdot & f1(rN) \\ & \cdot & \cdot & \cdot & \\ fN(r1) & \cdot & \cdot & \cdot & fN(rN) \end{vmatrix}.$$

Slater matrices provide basis states involving n particles and n orbitals. It is antisymmetric when exchanging any two particles.

Clifford algebra, over C, is a unital associated algebra A, involving a linear map f, from its underlying vector space V, into A. It is such that for v in V, there is a quadratic functional Q, where $f(v)^2 = Q(v)$ I, and I is the identity in A. These algebras are used in describing spin 1/2 particles in two-dimensional space. Additionally, they are useful in representations of quaternion vector spaces. Other representations of Clifford algebras involve quantization and lead to star algebras as well as extensions to a C* algebra. Importantly, a representation for the Fock space is that the Clifford algebra is a unital associative algebra generated by creation and annihilation elements on Z+1/2. Again subject to the usual Poisson bracket constraints. A very in-depth treatment of physics applications using Clifford algebras can be found in Borstnik and Nielsen (2021).

As mentioned earlier, Fock space representations F, with its dual space F*, exist for the Clifford algebra. These spaces both have unique vacuum states |0> and <0|. The vacuum states are also known as the Dirac sea (Rutgers Physics, 2018). As usual, fermionic states arise using the creation operators: $aj1^{\dagger} \ldots ajn^{\dagger}$ |0>.

15.5 Maya diagrams

Maya diagrams consist of a straight line pattern of black and white pebbles, dots, or other small objects located at every half-integer on the real line. Here, 0 denotes white and

X denotes black all at points Z+1/2. They are used as a free representation of fermionic Fock spaces. In Maya diagrams, there exists a negative integer m such that for all points j, less than or equal to m, white dots only occur. Similarly, there exists a positive integer n such that for all points j, greater than or equal to n, only black dots appear. In general, there also must be more than one transition of color. There is a 1−1 and onto correspondence between Maya diagrams and Young diagrams, see Appendix A.4. This correspondence involves critical parameters for both diagrams. For Maya diagrams, two parameters will be specified to start. These are two positive half-integer numbers w and b, w <b. That is, w and b are in Z+1/2, where w is the largest white dot location for which there does not exist any black dot located at a point less than w. Analogously, let b be the smallest location of a black dot for which no white dots have a larger location. Also needed is the total number nb of black dots that exist in the open interval (w, b); along with this, needed is the total number of white dots nw, in (w, b). An example of a Maya diagram follows.

Example 15.8:

Consider the following Maya diagram with 0 denoting a white dot and X for black dot:

...	−11/2.	−9/2..	−7/2..	−5/2..	−3/2..	−1/2..	1/2...	3/2...	5/2...	7/2	9/2..	11/2..	13/2..	15/2..	17/2...
...	0	X.	0.	X.	X.	X.	0.	X.	X.	0.	X.	X.	X.	0.	X...

In the aforementioned illustration of a Maya diagram, it follows that w = −11/2, since it is assumed that there are no black dots less than w. Also for a similar reason, b = 17/2. The number of black dots in the open interval (w, b) is nine, and the number of white dots in this interval is four.#

The bijection between these Maya and Young diagrams follows by utilizing the following constructive procedures. Let M stand for a Maya diagram and Y for a Young diagram, as given in Appendix A.4. As seen in the appendix, or observing Fig. 15.2, all Young diagrams are horizontal bar graphs. The bars are largest on the top row, and then they

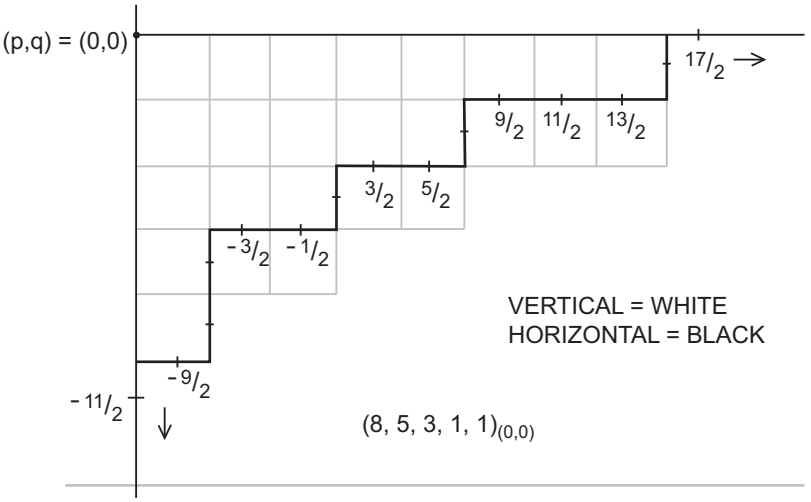


FIGURE 15.2 Young diagram obtained from the Maya diagram (see Example 15.9).

monotonically stay the same or decrease in length as the bars go downward. First it will be shown, given M, that the unique Y will be found. Correspondingly, the converse will be described. Starting from the Appendix A.4, Y is uniquely determined from $(n1', n2', \ldots, nk')_{(p,q)}$, where (p, q) are subscripts, providing left uppermost portion of the Young diagram. This is given as a bound matrix notation (Dourgherty and Giardina, 1987). Here, the positive integer nj′ is used to indicate the actual row length in Y. For the procedure given later, redundant rows will be eliminated. That is, now each row will have a unique size denoted by nj, j = 1, 2, . . . , r. Separate rules will be given to indicate when a row in Y is repeated. Assume that the M parameters, w, b, nw, and nb, specified earlier, are known. Using these quantities, p, q, and the nj will be found. To begin, p = (the integral part of b), minus nb. Similarly, q = (the ceiling function of w) with nw added. This locates the upper left-hand corner of the Y structure. Additionally, the first and largest block is given by n1 = nb. Subsequent blocks in Y are placed adjacent underneath if there are any.

When there are other blocks, they are found from M by noting the clusters of black dots and the number of separating white dots in the interval (w, b). A cluster of black dots consists of a contiguous string of black dots, and their lengths should be calculated, c1, c2, . . . cr, where c1 is the leftmost cluster and the others if any are in order. The white dots should also be noted. A single white dot separating two black clusters indicates the left cluster is of unique length. When k, k> 1, white dots separate two black clusters; this means that the block of cells in Y is repeated k times underneath each other. Also, note that the sum j = 1 to r of cj = n1. The second unique row in Y is found using n2 = n1−cr. Moreover, this row is repeated again if more than one white dot separates the clusters with cr and c(r − 1) black dots. In any case, the next unique row in Y is found using n3 = n2−c(r − 1). Again, check the number of white dots. In general, nk = n(k−1)−c(r−k+2), k = 2, 3, . . . , r. The ending condition is that the final unique block for Y is of size nr = n(r − 1)−c2 = c1. Finally, check to see the number of white dots in between the first two clusters of M; if there is only one white dot, then Y is complete. Otherwise, the final block with nr cells must be repeated k times; k is the number of white dots.

Example 15.9:

Consider the following Maya diagram with 0 denoting a white dot and X for black dot:

. . .	−11/2.	−9/2..	−7/2..	−5/2..	−3/2..	−1/2..	1/2. . .	3/2. . .	5/2. . .	7/2	9/2..	11/2..	13/2..	15/2..	17/2. . .
. . .	0	X.	0.	0.	X.	X.	0.	X.	X.	0.	X.	X.	X.	0.	X. . .

Assume also that there does not exist any other transitions in color. Accordingly, w = −11/2, since all white dots occur from minus infinity to and including w. Likewise, b = 17/2, since only black dots appear at and beyond this point. Finally nb = 8 and nw = 5; these are the number of black and white pebbles, always excluding the boundaries. From this information, a unique Young diagram will be created, see Fig. 15.2. Its starting point is (p, q), where p is the integer part of w that equals the integer part of 17/2 minus nb; thus p = 8−8 = 0. Also q equals the upper integer part of w = −11/2+nw; therefore, q = −5 +5 = 0. In this case, the Young diagram totally exists in the fourth quadrant and is

left justified on the negative part of the y-axis. The last comment means that all bars in Y have their left vertical boundary on the y-axis.

As can be seen from Fig. 15.2, the first block with n1 cells, hugging the x-axis, has n1 = nb = 8 cells long. The next lower block in Y is of unique length n2, since the last cluster c4 of M is separated by a single white dot; consequently n2 = n1−c4 = 8−3 = 5. Similarly, since a single dot separates the second and third clusters, this implies n3 in Y is unique and it is n3 = n2−c3 = 5−2 = 3. However, since two dots separate the left, first, and second clusters in M, the blocks n4 and n5 are equal and of length n3−c2 = 3−2 = 1. This agrees with the length of the bottom-most block in Y and the length of the first cluster in M.#

Now going the other way, given a Young diagram, the corresponding unique Maya diagram will be constructed. The procedure is given after an initialization or a setting up. Again, consult Appendix A.4. M is described by (p, q), its integer location in R^2, along with the length nj, of every block, each located adjacent to and underneath each other, left justified. Recall the blocks top down form a finite nonincreasing sequence of lengths. Here, nj can equal n(j+1), that is, the block lengths can be repeated underneath each other. It is useful to construct ruled horizontal and vertical lines. These lines are orthogonal to each other meeting at (p, q). Ruled markings increase in the horizontal direction and decrease in the vertical direction all by the same units, the size of a cell edge. The Y structure is left justified and are usually located in the fourth quadrant. All rulings are compatible with the cell size making up each bar. Intuitively, all vertical line segments in Y symbolize white dots in M, and horizontal line segments in Y denote black dots.

The setup is crucial; it provides the environment for the initial, recursive, and final stages in determining M. The largest location of a white point w in M for which all smaller half-unit points are also white is easy to find. It is found at half a unit, in the vertical direction, right below the bottom bar of Y. So, the leftmost cell of the bottom bar in Y is (p, p + 1] x [− h, − h + 1); then w = − h −1/2. Here h is the total number of bars constituting Y; it is the number of nj's. Place white dots at w and at all half-integer points to the left of w in M.

Next, the profile of Y is traversed, from left to right and bottom up. This is the lower and left-hand boundary line segments. Black dots are placed in M for each horizontal line segment of every cell in Y traversed along the profile. White dots are placed in M, for every vertical cell wall in the profile when going upward. The location in M for each of these dots starts with w and is found accumulatively, that is, by adding one black dot to the location in M for every line segment of each cell going to the right. Also add one white dot for each line segment going up in the profile of Y. Accordingly, dots will be issued to locations w+1, w+2, and so on. The first black dot is found traversing the bottommost and leftmost cell (p, p+1] x [− h,−h+1), going horizontal, and thereby marking a black dot at w+1 = 1/2−h. The next dot is placed at w+2; it will be a black dot by going horizontal if the bottom bar has more than one cell, but if the bottom block has one cell only, up is the direction and a white dot is placed at w+2. The next dot is placed at w+3; it is black for horizontal movement and white for vertical movement per each unit line segment traversed in the profile. The final line segment in the profile must be vertical. It corresponds to the right-hand

vertical edge of cell (p+n1−1, p+n1] x [q−1, q), indicating a white dot should be placed at p+n1−1/2. Finally, b = p+n1+1/2, and at b and all larger half-integer points, there should be black dots.

Example 15.10:

Consider the Young diagram illustrated in Fig. 15.3. From this figure, it is seen that the Y structure is located at (p, q) = (0, 1), and the length of the bars is given by the finite sequence: (n1, n2, n3, n4) = (5, 3, 3, 1), and h = 4 since there exist 4 bars in Y. As a bound vector, Y is $(5,3,3,1)_{(0,1)}$.

Going half a unit under the lowest bar gives w = −3 1/2. This value could be found algebraically by using w = q−h −1/2 = 1−4−1/2. That is, it can be found from the bound vector, whose length is four, and starts at (0, 1). The y component is one. In any case, this point should be marked white along with all half-integer values less than w. So:

11/2.	−9/2..	−7/2..	−5/2..	−3/2..	−1/2..	1/2...	3/2...	5/2...	7/2...	9/2..	11/2..	13/2..	15/2..	17/2...
0	0	0.												

Next, the bottom bar consisting of a single cell is encountered; it is horizontal, so a black dot is drawn at position −5/2, one unit more than w. A single black dot appears since the last tuple in the bound vector is one. Go up on the bottom bar on the profile, of Y, producing a white dot at −3/2. Now the next to the bottom bar is encountered in a horizontal manner. Since there are two cells involved, two black dots are issued at points −1/2 and 1/2. These two black dots appear since the difference between the next to last tuple and the last tuple in the bound vector is two. Consequently, at this point, the M diagram is as follows:

11/2.	−9/2..	−7/2..	−5/2..	−3/2..	−1/2..	1/2...	3/2...	5/2...	7/2...	9/2..	11/2..	13/2..	15/2..	17/2...
0	0.	0.	X.	0.	X.	X								

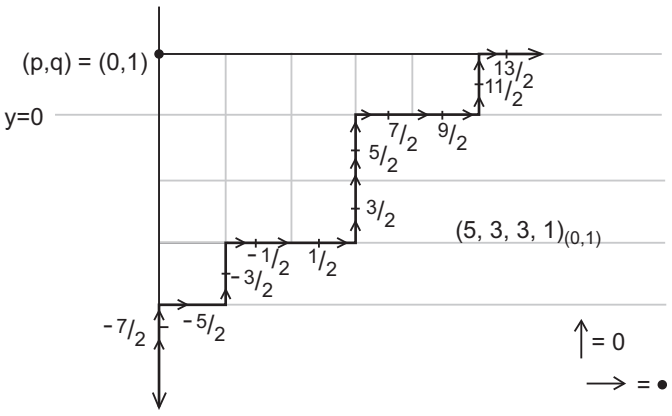


FIGURE 15.3 Young diagram to find Maya diagram.

Going up in the profile for blocks 3 and 2 induces white dots at locations 3/2 and 5/2. Again, these two consecutive white dots could be seen from the bound vector since the two tuples have the same number, three. Then going horizontal on block 1 yields black dots at 7/2 and 9/2. As before, using the difference between the first and second tuple provides the number of black dots. Finally, the profile traversing ends by providing a white dot at 11/2. Starting from 13/2 inclusive, all black dots exist. The final M structure is:

11/2.	−9/2..	−7/2..	−5/2..	−3/2..	−1/2..	1/2...	3/2...	5/2...	7/2...	9/2..	11/2..	13/2..	15/2..	17/2...	
0	0.	0.	X.	0	X	X	0.	0.	X.	X.	0	X.	X	X	.#

15.6 Maya diagram representation of fermionic Fock space

Fermionic Fock space as a vector space V = direct sum complex-valued basis indexed by the half-integers. For an element w, in this space it can be expanded in a basis: w = ej1 & ej2 & ..., where jk is a strictly increasing sequence of half-integers. Also, & is the wedge, or and type product. Each wedge can be visualized as a Maya diagram. Moreover, fermionic Fock space can be viewed as the free span over C of all Maya diagrams. The standard basis for this space is the Maya diagrams. The vacuum state (vac) has white dots exclusively at every positive half interval. In general, if there exist black dots only beyond some half-integer n, the diagram is said to have a vacuum at level n. This is also called the Fermi level.

For a fixed Maya diagram M, a k translation operator moves black dots to the left k units. It does this, provided that there does not exist a black dot presently occupying the half-integer position in M. Specifically, for the translation operator sk = sum, $\Sigma(-1)^{j(M,M')}$. The translation ends in the Maya diagram M′ and involves all Maya diagrams continually starting from M. It results in all diagrams for which a black dot moves k spaces to the left with a plus or minus sign. The function j(M, M′) indicates the number of black dots bypassed in M, when the available black dot relocates. The translation operation can be expressed as a convolution-type operator involving the creation $f^{\dagger}$ and annihilation operation f. Indeed, sk equals the following sum, $sk = \sum_{ninZ+1/2} f^{\dagger}(k+n)f(n)$. Again, however, the function j must be utilized to keep track of the number of dots jumped over (Bouttier, 2019). For the next example, use v(a/2), where a is an integer as a representation of a black dot. These are the dots that will relocate to the left.

Example 15.11:

In this example, the original standard basis is given by the Maya diagram M, specified below. It is given involving wedge products, v(−5/2) & v(−1/2) & v(1/2) & v(7/2) & v(9/2) & v(13/2) .... As mentioned earlier, each v(a/2) indicates a black dot location. All other locations can be considered blank or white dots. Say that a translate minus three units is to be found, that is, s(−3) M. It is useful to provide a list of black dot locations in M that are relocatable, that is, they do not contain a black dot, this list is fixed. Also a corresponding list for the value of j(M,M′) should be created. The first list is {−5,2, −1/2, 9/2, 17/2}.

These are the four black dots that will translate to the left. The transition is performed one operation at a time using this list, left to right. Note that the black dot at location 1/2 is not relocatable because -3 unit away, to the left, is occupied by the black dot at $-5/2$. The corresponding ordered list for the number of jumped black dots j(M,M′) is {0, 1, 1, 2}. The last number in this list 2 is the number of black dots in M, between 11/2 and 17/2. This is the number of black dots to jump over when the black dot at 17/2 relocates three units to the left. For this basis M, the resulting basis M′ will be the sum or difference of four lower diagrams. Immediately below M is the translation of v($-5/2$); it relocates three units to the left, arriving at $-11/2$. The resulting Maya diagram, say M + , is positive since j(M,M +) = 0. The next translation is of v($-1/2$), and the Maya diagram appears underneath M + . However, it has a negative sign, since in the translation this black dot had to jump over one other black dot in M to arrive at $-7/2$. M′ is the sum of the bottom four Maya diagrams:

M:	11/2.	–9/2..	–7/2..	–5/2..	–3/2..	–1/2..	1/2…	3/2…	5/2…	7/2	9/2..	11/2..	13/2..	15/2..	17/2…
	0	0.	0.	X.	0	X	X	0.	0.	X.	X.	0	X.	X	X
M':	–11/2.	–9/2..	–7/2..	–5/2..	–3/2..	–1/2..	1/2…	3/2…	5/2…	7/2	9/2..	11/2..	13/2..	15/2..	17/2…
	X	0.	0.	0.	0	X	X	0.	0.	X.	X.	0	X.	X	X
–	–11/2.	–9/2..	–7/2..	–5/2..	–3/2..	–1/2..	1/2…	3/2…	5/2…	7/2	9/2..	11/2..	13/2..	15/2..	17/2…
	0	0.	X.	X.	0	0	X	0.	0.	X.	X.	0	X.	X	X
–	–11/2.	–9/2..	–7/2..	–5/2..	–3/2..	–1/2..	1/2…	3/2…	5/2…	7/2	9/2..	11/2..	13/2..	15/2..	17/2…
	0	0.	0.	X.	0	X	X	X.	0.	X.	0.	0	X.	X	X
+	–11/2.	–9/2..	–7/2..	–5/2..	–3/2..	–1/2..	1/2…	3/2…	5/2…	7/2	9/2..	11/2..	13/2..	15/2..	17/2…
	0	0.	0.	X.	0	X	X	0.	0.	X.	X.	X	X.	X	0 .

It is interesting to use the creation and annihilation formula in a convolutional manner, $sk = \sum_{n in Z+1/2} f^{\dagger}(k+n)f(n)$, to perform the translations given earlier. In this example, $k = -3$. Accordingly, s(-3) equals the sum for n in Z+1/2 of $f^{\dagger}(n-3)$ f(n). Starting with M process will provide the same M′ as mentioned earlier. Begin with the first available black dot X at its initial location $-5/2$ in M. The annihilation operation f removes the black dot at $-5/2$; this is f($-5/2$). This black dot was available because the relocation position $-3+(-5/2) = -11/2$ is not occupied by a black dot. Also jMM′ = 0, since there are no black dots in between the present and relocatable position. Accordingly, a plus sign is used when the creation operation $f^{\dagger}(-11/2)$ is employed, and it relocates the starting black dot.

The next black dot X on the relocatable list is at $n = -1/2$ in M. In this case, this point gets annihilated by f($-1/2$), thus removing the black dot and replacing it with nothing, that is, a white dot. Next, location -3 1/2 in M was inspected to be unoccupied, and since this is the case, the creation operator, then $f^{\dagger}(-7/12)$, puts a dot X at this location in M′. However, in this situation one dot was jumped over in the relocation process, causing a minus sign to be included in M′. Similarly, for the black dot X at 9/2, f(9/2) performs an annihilation of X and f*(3/2) creates a new black dot at 3/2 with an overall minus sign since one black dot was jumped over. Finally, the back dot at 17/2 is first annihilated and a new black dot at 11/2 is created, but this time with a plus sign since two black dots were jumped over in the process. The result using the convolution is the same as before.#

15.7 Young diagrams representing quantum particles

For systems of n identical particles with permutation invariance for the Hamiltonian H, then the eigenfunctions of distinct permutations share their same eigenvalue E. Let P be a permutation operator and e(x1, x2, ... xn) be an eigenfunction with eigenvalue E. Then Pe = Ee. The symmetrizing operation is S = (1/n!) sum P, and the antisymmetrizing operation is A = (1/n!) sum P dp; here, all sums are over Sn, the permutation group of n objects. The parity dp is 1 for P even and −1 for P odd. Note that $dp = (-1)^t$ where t is the number of transpositions in P; also $dp=(-1)^{(n-c)}$, where P consists of c cycles. Boson states are represented by a single row of the Young tableau where the states are completely symmetric. Likewise, Fermi states are represented by single columns in the Young tableau where the states are completely antisymmetric (Yong, 2007). Young tableaux are Young diagrams with markings within cells to keep track of configurations.

Example 15.12:

Let u and v be orthogonal single-particle states, and w the two-particle state, w = u v. More precisely, w = u(x1) v(x2). From above, for two particles, the symmetrized operation, S12 = 1/2 (I+P12), where I is the identity operator in the symmetric group S2. For w, this operator projects the symmetric part; thus: S12w = 1/2 [u(x1) v(x2) + v(x1) u(x2)]. The antisymmetrized operation is A12 = 1/2 (I−P12), and so A12w = 1/2 [u(x1) v(x2) −v(x1) u(x2)].

The Young diagrams associated with these particle configurations contain two cells. These diagrams have adjacent two horizontal cells or two cells one above the other. The former diagram corresponds to the class of two one-cycle symmetric states and the latter to a two cycle that is a transposition and therefore antisymmetric state. The two horizontal cells correspond to any of the states: u(x1) u(x2), v(x1) v(x2), 1/2[u(x1) v (x2)+v(x1) u(x2)]. The last quantity can be written as 1/2 [u1 v2+v1 u2]. While the vertical bar corresponds to the state 1/2 [u(x1) v(x2)−v(x1) u(x2)], this can also be written as 1/2 [u1 v2−v2 u1].#

Example 15.13:

For the situation of three-particle states, the construction of Young's diagrams becomes somewhat more complex. The arrangement of u, v, and z, three single-particle states, will be utilized. There exist six linearly independent functions, which will be described in terms of symmetry properties. First, the completely symmetric pattern w = 1/3! [uvw+vwu+wuv+vuw+uwv+vuw]. This again corresponds to three 1 cycles and is a single horizontal bar in V. It again is the identity representation. The completely antisymmetric configuration is w = 1/3! [uvw−wvu+wuv−uwv+vwu−vuw]. In Y, this exists as a single column. Moreover, this generates another irreducible representation for S3.#

Four more functions besides the bosonic and fermion exist in the three-state quantum particle environment. These have mixed symmetry; first, two particles can be

symmetrized, and then two particles can be antisymmetrized. These particles are known as quark-flavor eigenstates. The Y description for the four quark states always has a top bar of two cells and a bottom bar of a single cell. In terms of the quark flavor, the generation of wave functions for baryons can be illustrated, leading to a 27-dimensional representation of SU(3). Here, the Clebsch-Gordon series could be found consisting of the SU(3) irreducible representations. They can be described in terms of Y and Weyl diagrams and occur in angular momentum coupling (Messiah, 1981).

15.8 Bogoliubov transform

The Bogoliubov transform is an isomorphism for the CCR or the CAR. Specifically, the transforms provide conditions for which the creations and annihilation operations can be transformed into a new set of equivalence operations. Let the original creation and annihilation operations be $f^\dagger$ and f, respectively. The new operations are $a^\dagger$ and a. The Bogoliubov transform T is such that $(a^\dagger\ a)' = T\ (f^\dagger\ f)'$, where T is such that for A =

$$\begin{matrix} |0 & 1| \\ |1 & 0| \end{matrix}$$

$$TAT' = A.$$

Example 15.14:

Let T be given below along with T, T A, as well as T′, in the following order:

$$\begin{matrix} |i & 0| & |0 & i| & |i & 0| \\ |0 & -i| & |-i & 0| & |0 & -i| \end{matrix}$$

The product T A T′ = A. As a consequence, in this case the new creation and annihilation operations are given from the original ones by $a^\dagger = i\ f^\dagger$, a = −i f.#

15.9 Parafermionic and parabosonic spaces

These spaces are built on the premise that irreducible representations of the permutation group govern the transformation of wave functions. In particular, particle states transform under generalized symmetry. The states exist in between bosonic and fermionic states. Precise representations are given using Young diagrams. Quons are a theoretical particle believed to exhibit symmetries as well as antisymmetries. An algebra for working with quons involves operators from the bosonic as well as the fermionic Fock spaces. Specifically, the operator is a convex combination of the Lie bracket and the Poisson bracket. Indeed, for r in the interval [−1, 1], the operator involves both the annihilation a and creation operation $a^\dagger$ and is (1+r)/2 [am, $an^\dagger$]+(1−r)/2 {am, $an^\dagger$} = dmn, where dmn = 0, unless m = n; then it equals one. The vacuum state is defined as in Fock space and am|0> = 0 as before. For r = 1, the bosonic Fock

space is described in the aforementioned formula. Similarly for r = −1, the fermionic Fock space operations are applicable. Irreducible representations of the symmetric group are weighted in describing the states for quons.

15.10 Segal−Bargmann−Fock operations

An application of holomorphic functions in quantum mechanics is in describing the phase space for a quantum particle moving in R^n. Here the Segal−Bargmann−Fock (SBF) space is employed (Guillemin, 1984). It is a Hilbert space, with an inner product developed so that creation aj† and annihilation operators ak commute among themselves, and such that aj† is the adjoint of aj. The inner product of two holomorphic functions F and G in C^n is for functions in $L^2(C^n)$ and is <F, G>, which equals the integral, $\int(F(z)^*G(z)e^{-|z|^2})dz$ where dz is the 2n-dimensional Lebesgue measure on C^n. The representation is a Gaussian-type integral.

Fock observed that conjugate coordinate operations in complex variables obey the commutation relations. Accordingly, the conjugate coordinate operations are a form of the Cauchy Riemann criteria for analyticity. These equations involve partial derivatives: $\partial/\partial zj^* = 1/2\ (\partial/\partial xj + i\partial/\partial yj)$ and $\partial/\partial zj = 1/2\ (\partial/\partial xj - i\partial/\partial yj)$. A holomorphic function can never be a function of z*, so the partial derivative of any analytic function F is such that $\partial F/\partial zj^* = 0$. Forgetting about Planck's constant and identifying zj with multiplication-type operation and the partial derivative $\partial/\partial zk$ with the momentum operation, then the commutators: $[zj, zk] = [\partial/\partial zj, \partial/\partial zk] = 0$, and $[\partial/d\partial zj, zk] = 0$ for j not equal to k, and one when they are equal. Here, aj* is the raising or creating operation, and aj is the lowering or annihilating operation with the domain of holomorphic polynomials, which is a dense subspace of SBF.

15.11 Many-body systems and the Landau many-body expansion

Second quantization provides the foundation for many-body quantum systems. Fock space provides a basic setting along with bosonic and fermionic creation and annihilation operations. Representations are given for one-body operators. These operators manipulate single particles in N-dimensional Hilbert space. Similarly, two-body operators are described, which illustrate the interaction between particles.

Landau presented an expansion involving the number of interacting particles produced by operators of various arity with interacting arguments. For operators of arity one, also called single-body or single-particle operators, they are denoted by T1. The operators of arity two whose operands interact with each other are called two-particle or two-body operators and are denoted by T2. The operators of arity three whose operands interact with each other are called three-particle or three-body operators and are denoted by T3. And so on.

The expansion is as follows: Sum, $\sum_j$ T1j + Sum, $\sum_{j<k}$ T2jk + Sum, $\sum_{j<k<m}$ T3jkm + . . .

In second quantization, this expansion involves the bosonic or fermionic creation operators a† and c†, as well as the annihilation operators a and c, respectively. In this case, the expansion is as follows: Sum, $\sum_{j,k}$ T1aj†ak) + 1/2(Sum, $\sum_{j,k,m,n}$ T2aj†ak†an am) + . . . This

expansion was used for bosons, but the same expansion holds for fermions. The creation and annihilation operations are given next with applications to the one-body and two-body problem in second quantum.

15.12 Single-body operations

Single-particle or one-body operators are defined in N particle Fock space as summation, $\sum_{k=1}^{N}$ bk, where bk is an operator acting on the kth particle state. Of principal importance is the number operator nk, which is the number of particles in location k. The quantity nk = $ak^{\dagger}ak$; here it is given for bosons. Recall the eigenstate $(ak^{\dagger})^{k}$ |vac> of the number operator has eigenvalue k, that is, nk $(ak^{\dagger})^{k}$ |vac> = k $(ak^{\dagger})^{k}$ |vac>. Moreover, the creation operator $aj^{\dagger}$ (|..., nj, ...>) = $(nj+1)^{1/2}$ (|..., nj+1, ...>), and the annihilator operator is such that aj |..., nj, ...> = $nj^{1/2}$ |..., nj−1,...,>. For the fermionic Fock space, the number operator is nk = $ck^{\dagger}ck$. The creation operator this time is $cj^{\dagger}$(|..., nj, ...>) = $(1-nj)^{1/2}$ (|..., 1−nj, ...>), and annihilation operator is such that cj (|..., nj, ...>) = $(nj)^{1/2}$ (|..., 1−nj, ...>). The large difference between bosonic and fermion operators is that the former is symmetric in the creation of two particles, and the latter is antisymmetric. That is, $aj^{\dagger}ak^{\dagger}|0> = ak^{\dagger}aj^{\dagger}|0> = |1j\ 1k>$, but $cj^{\dagger}ck^{\dagger}|0> = -\ ck^{\dagger}cj^{\dagger}|0> = |1j\ 1k> = -\ |1k\ 1j>$.

One-body operation T1 transfers a particle from state v to state v′ with probability <v |T1| v′>, where T1 = sum, $\sum_{j=1}^{N}$ Tj, and Tj acts only on particle j. The formula is T1 = sum, $\sum_{v,v'} <v|T|v'> av^{*}av'$.

Example 15.15:

The kinetic energy of a system of N particles is T, the sum, $\sum_{j=1}^{N} pj^{2}/(2m)$, with pjs the momentum and m the particle mass. The second quantization for the single-body operation is T1 = sum, $\sum_{u,v} <u|p^{2}/(2m)|v> au^{\dagger}av$. This formula becomes T1 = sum, $\sum_{k,s} h^{2}k^{2}/(2m)aks^{\dagger}aks$. Here k denotes the momentum vector, and s is the spin. The notation is summarized by noting that the number operator, in this case, is nks = $aks^{\dagger}$ aks, which counts the number of particles with momentum k and spin s.

15.13 Two-body operations

Similar to one-body operators, two-particle operators conserve the number of particles.

The formula for two-body interaction is T2 = 1/2 sum for v, v′, u, and u′ of <u, v |T| u′, v′> $au^{\dagger}av^{\dagger}av'au'$. The quantity <u, v |T| u′,v′> represents a two-particle integral involving T (Hall, 2013).

References

Borstnik, N., Nielsen, H., 2021. How Does Clifford Algebra Show the Way to the Second Quantized Fermions With Unified Spins, Charges and Families, and With Vector and Scalar Gauge Fields Beyond the Standard Model. Department of Physics, University of Ljubljana.

Bouttier, J., 2019. Fermions in Combinatorics: Random Permutations and Partitions. Available from fermions.sciencesconf.org.
Dourgherty, E., Giardina, C., 1987. 0-13-453283-X Image Processing Continuous to Discrete. Prentice-Hall.
Gazeau, J., 2009. Coherent States in Quantum Physics. Wiley.
Guillemin, V., 1984. Toeplitz operators in n-dimensions. Int. Eq. Op. Theory 7.
Hall, B., 2013. Quantum Theory for Mathematicians. Springer.
Messiah, A., 1981. Quantum Mechanics, Vol. 1. North Holland, ISBN 10: 0720400449.
Rutgers Physics, 2018. Chapter 10, Clifford algebras.
Slater, J., 1929. The theory of complex spectra. Phys. Rev. 34 (2).
Townsend, J., 2000. 1-891389-13-0 A Modern Approach to Quantum Mechanics. University Science Books.
Yong, A., 2007. What is a Young Tableau. Notices AMS 54 (2).

This page intentionally left blank

CHAPTER

16

Underlying theory for quantum computing

16.1 Quantum computing and quantum circuits

Quantum computing can be conducted similarly to conventional computing involving circuits with inputs and outputs. It can also be conducted as a specialized optimization type, namely adiabatic computing. In any case, all methods mentioned in this or the following chapters have a speedup greater than that of a conventional computer. However, the speedup of quantum computation over a conventional computation of order N can never be more than quadratic. It was shown (Bennett et al., 1997) that the quantum computation of the order square root of N is as optimal as possible. Quantum circuit-type computers and conventional computers are similar in some respects; however, they do differ in many ways.

The most fundamental thing about quantum circuits is that the inputs must be qubits from some Hilbert space. Additionally, these circuits are described similarly to input-output block-type diagrams as in engineering disciplines. In engineering, the different inputs are usually not linked together; they have no relation to each other. In quantum, however, these diagrams usually symbolize input qubits that are tensored together, and they enter the circuit, often linked in this structural manner. Moreover, the output of quantum circuits is always equal in number to the number of inputs. Usually, computation ends with a measurement of a tensor product of several output qubits. The result is a collapse of the wave function. This will be the case in all quantum algorithms described later. Also, all the operations T performed on the qubits must be linear unitary operations. The latter requirement is $T\ T^* = T^*T = I$. The set of all unitary operators mapping a Hilbert space of qubits into itself forms a group, called the Hilbert group. The MSA description of a Hilbert group is provided in Fig. 16.1. Here, as usual, the polyadic graph illustrates the closure operations from the signature sets. Accompanying is a list of equational identities needed for the rigorous specification of a group.

The simplest way of showing that an operator T is unitary in a quantum circuit is to show that the product of the matrix with its adjoint is the identity matrix. A sufficient condition for T to be a unitary operator is if it is both an involution and it is normal. That is, show that $T^2 = I$ and $T\ T^* = T^*T$, respectively. This result is a consequence of the spectral theorem for finite-dimensional normal operators. If T is normal, then $T = A\ D\ A^*$, where D is a diagonal matrix consisting of eigenvalues of T with multiples repeated, and A is

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00006-0

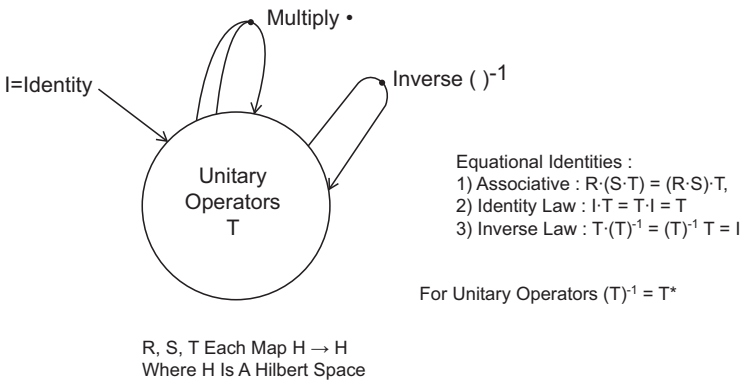


FIGURE 16.1 Hilbert group.

unitary. If T is an involution, then $D^2 = I$, so all the eigenvalues are plus or minus one. This shows that D is both self-adjoint and unitary, because $T = A\ D\ A^*$ with A unitary shows that T is unitary also. The ket is represented as a unique point on the Bloch sphere.

Example 16.1:

Consider the carrier set to be C^2, and the operator T: $C^2 \rightarrow C^2$. Note that if $T^2 = I$, that is, if T is an involution, then this is not enough to verify that T is unitary. Just to show that an operation is reversible is not enough to prove the operation is unitary. To see this, let the involutionary matrix T =

$$\begin{vmatrix} -1 & 1 \\ 0 & 1 \end{vmatrix}$$

Then T is not a unitary matrix even though $T^2 = I$, because T T* does not equal I.#

16.2 Single-qubit quantum gates

Unitary operations in a quantum circuit are called quantum gates. Thus, a quantum circuit is comprised of quantum gates. Quantum gates usually have a single-qubit input, but some gates have multiple-qubit inputs. When a single qubit is the input to a gate, this is most often performed using a representation in C^2, as in the aforementioned example. The following examples again involve linear operators acting on qubits represented in C^2. All these operators belong to the Hilbert group mentioned earlier, but in addition all of these operators also belong to specific Lie groups. As a reminder, important algebraic specifications of some Lie groups will be quickly reviewed. In particular, the special unitary Lie group SU(n,C) is a real Lie group consisting of unitary n by n matrices with determinant one. The general unitary real Lie group U(n,C) consists of n by n unitary matrices with determinant absolute value equaling one. In general, the global phase of the qubit $|v>$ is ignored. More rigorously, qubits belong to the projective space $P^n(C) = CP^n$ but this fact will not be emphasized. In any case, for p a real number in $[0, 2\pi)$ and t real-valued in $[0, \pi]$, the general ket in C^2 is $|v>$, and it

can be written using a superposition of kets: $|v> = (\cos(t/2))|0> + e^{ip} \sin(t/2)|1>$. Only the global phase is left out in the previous representation.

Example 16.2:

The Hadamard gate H has a single qubit as input, say $|v> = a|0> + b|1> = (a\ b)'$, in C^2, where $|a|^2 + |b|^2 = 1$. This gate is specified by the matrix H given by

$$1/2^{1/2} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

This matrix is an involution, that is, $H^2 = I$. Inspection shows that it is symmetric and therefore self-adjoint, and consequently it is normal. Accordingly, H is also unitary. The multiplication $H|v>$ gives the column vector, $1/2^{1/2}$ $(a+b\ a-b)'$. Utilizing ket notation again, it follows that $H|v> = 1/2^{1/2}[(a+b)|0> + (a-b)|1>]$. In particular, $H|0> = 1/2^{1/2}[|0> + |1>] = |+>$ and is called the plus state. Similarly, $H|1> = 1/2^{1/2}[|0> - |1>] = |->$ and is called the minus state. These two qubits lie on the x-axis on the surface of the Bloch sphere, see Figure 5.3.

A different representation for the Hadamard matrix is using outer products involving ket and bra qubits; indeed, $H = 1/2^{1/2}(|0><0| + |0><1| + |1><0| - |1><1|)$. This representation is useful because there is no need to use vectors in C^2 as a representation. In short, kets and bras could be manipulated directly by H described earlier. Illustrations follow on how to evaluate H acting on an input qubit. For instance, to find the Hadamard matrix operating on the south pole qubit, $H|1>$. Use linearity to obtain $H|1> = 1/2^{1/2}(|0><0|1> + |0><1|1> + |1><0|1> - |1><1|1>) = 1/2^{1/2}(|0> - |1>)$. The last result follows since $|0>$ and $|1>$ are each of unit length and are orthogonal to each other, so, for instance, $<0|1> = 0$ and $<0|0> = 1$. Finally, note that the determinant of H is minus one. This shows that H is in the Lie group U(2, C), but it is not in the Lie group SU (2, C). However, modified Hadamard matrices are often utilized, for instance, iH is employed. In this case, the matrix iH is in SU(2,C), since (iH) (iH)* = (iH) (- iH) = $H^2 = I$.#

Example 16.3:

The NOT gate also has a single-qubit input; in this situation, the output just exchanges zero qubits for one qubits and vice versa. Thus, for the qubit $|v>$ as in the previous example, $|v> = a|0> + b|1> = (a\ b)'$, in C^2, where $|a|^2 + |b|^2 = 1$. Evaluation using NOT $|v> = \text{NOT}(a|0>) + \text{NOT}(b|1>) = a\ \text{NOT}|0> + b\ \text{NOT}|1> = a|1> + b|0>$, just illustrated is the linearity property of this operator. Also shown is the use of parenthesis to gain more clarity, for instance, use NOT$(a|0>)$, but don't use NOT$a|0>$. In any case, operating in the Hilbert space C^2, the matrix for the NOT operation is as follows:

$$\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$$

This matrix is both symmetric and an involution, and so NOT is a unitary operation. NOT gate can be described as linear combination of outer products. Thus, NOT = $(|0><1| + |1><0|)$. Since the determinant of NOT is minus one, again like the Hadamard matrix, NOT is in U(2, C) and it isn't in SU(2, C).#

Example 16.4:

The identity gate I, where I: $C^2 \rightarrow C^2$, is given as follows:

$$\begin{matrix} |1 & 0| \\ |0 & 1| \end{matrix}$$

It is represented in terms of outer product bras and kets by $I = |0><0| + \{1><1|$. As such, any input $|v>$ to this gate exits unaltered. Since the determinant is one, I is in SU(2,C).#

The next example of a single-qubit operation involves a single, but famous phase gate.

Example 16.5:

The $\pi/8$ phase gate also known as the T operation is part of the Hadamard, H, T gate set. The corresponding matrix operating in C^2 is as follows:

$$\begin{matrix} |e^{-i\pi/8} & 0| \\ |0 & e^{i\pi/8}| \end{matrix}$$

Note that this matrix is not an involution. However, of importance is that the T matrix is a unitary matrix and the determinant is one. Accordingly, T is a member of the Lie group SU(2, C). Often, in quantum circles, the T gate is defined slightly different. Here, it is represented as the above T gate multiplied by $e^{i\pi/8}$, and sometimes additionally as follows:

$$\begin{matrix} |i & 0| \\ |0 & e^{i\pi/4}|.\# \end{matrix}$$

Example 16.6:

A generalization of the T gate above is the PHI gate, which involves a nonzero angle of ϕ, with an uncountably infinite number of such values for this angle. Specifically, PHI =

$$\begin{matrix} |1 & 0| \\ |0 & e^{i\phi}|.\# \end{matrix}$$

Use of an operator having the versatility of employing an uncountable number of angles is a must in quantum computing. This is because the Lie groups involved are topological spaces with an uncountable number of elements. The next quantum gate S is a member of the Clifford+T set. The operator T is illustrated in Example 16.5.

Example 16.7:

The S gate is called the phase-shift gate. It is represented as a matrix mapping Hilbert space C^2 into itself. S =

$$\begin{matrix} |1 & 0| \\ |0 & i| \end{matrix}$$

This matrix is unitary, but the determinant equals i; accordingly, S is a member of the unitary Lie group U(2, C). It is not in SU(2, C). Note that this is a special case of the PHI gate specified above with $\phi = \pi/2$. This operator along with the T operator, as well as H, forms the Clifford+T finite group. The Clifford single-qubit group is described in a subsequent section. It was shown that the Clifford + T set is universal and certain strings of matrices from this set can be represented by matrices with algebraic number entries. Also, certain approximations to entries U in SU(2, C) in terms of H, S, and T are strings of matrices of the form: {T or empty} {HT or HST}.#

Example 16.8:

This is an example of how the S gate operates on the qubit, $|q> = (\cos(t/2)\ e^{ip} \sin(t/2))'$, on the Bloch sphere. First form $S|q> = (\cos(t/2)\ ie^{ip} \sin(t/2))$; then since $i = e^{i\pi/2}$, it follows that $S|q> = (\cos(t/2)e^{i(p+\pi/2)}\sin(t/2))$. Referring to the Bloch sphere, the new qubit after the application of S is now located at a new longitude, but the same latitude as $|q>$. It is located 90 degrees rotated counterclockwise, about the z-axis, away from $|q>$. The z-axis is pointing up. The Darboux vector is the name often used for the angular velocity vector causing the rotation.#

16.3 Pauli rotational operators

The next gate set involves the three Pauli matrices that were previously specified, and again given below, in order X, Y, and Z:

$$\begin{matrix} |0 & 1| & |0 & -i| & |1 & 0| \\ |1 & 0| & |i & 0| & |0 & -1|. \end{matrix}$$

These matrices are exponentiated, resulting in the Pauli rotation operators. They are illustrated in the next example. All rotation operators can be found using Frobenius covariants with Lagrange-Sylvester interpolation, or by using power series methods. See Section 5.5, for the Frobenius covariants with Lagrange-Sylvester interpolation.

Example 16.9:

The three rotation matrices corresponding to the Pauli matrices are as follows: $R_X(t) = e^{-itX/2}$, $R_Y(t) = e^{-itY/2}$, and $R_Z(t) = e^{-itZ/2}$. These rotation matrices are presented in order as follows:

$$\begin{matrix} |c\ t/2 & -is\ t/2| & |c\ t/2 & -s\ t/2| & |e^{-it/2} & 0| \\ |-is\ t/2 & c\ t/2| & |s\ t/2 & c\ t/2| & |0 & e^{-it/2}| \end{matrix}$$

Here, c t/2 stands for cos(t/2) and s t/2 represents sin(t/2). For a single qubit for which these matrices are applied, they rotate the qubit about the corresponding x-, y-, or z-axis of the Bloch sphere. All three matrices are in SU(2, C).

To derive $R_X(t) = e^{-itX/2}$, note that $R_X(t) = ct/2I -$ is $t/2X$, where I is the identity 2 by 2 matrix and X is the first Pauli matrix. From the cosine and sine expression, the result follows immediately by substitution. More generally, the expression for $R_X(t)$ can again be found as indicated earlier by using the eigenvalues of X. This is the method specified in Section 5.5. Utilizing the Frobenius covariant method, since the eigenvalues of X are $x1 = 1$ and $x2 = -1$, the Frobenius covariants are $X1 = 1/2$ (X+I) and $X2 = -1/2$ (X − I). The Lagrange-Sylvester interpolation formula for $R_X(t) = e^{-itX/2} = e^{-ti/2}X1 + e^{ti/2}X2 = 1/2[(e^{-ti/2} + e^{ti/2})I + (e^{-ti/2} - e^{ti/2})X]$. Substituting in I and X yields the same solution as the one previously given.#

Of importance is the meaning of the angle t in the Pauli rotation matrices. This angle t in R(t) is the angle employed according to the Bloch sphere spherical coordinate system. An example follows.

Example 16.10:

On the surface of the Bloch sphere, the spherical coordinate vector q locates the pure qubit, $|q>$. For instance, along the x-axis $q_x = (1\ 0\ 0)$ is the location for the qubit, $|q_x> = 1/2^{1/2}\ (|0> + |1>) = (\cos(\pi/4)\ \sin(\pi/4))' = 1/2^{1/2}\ (1\ 1)'$. Also, along the x-axis, note that $q_{-x} = (-1\ 0\ 0)$ is the location for the qubit, $|q_{-x}> = 1/2^{1/2}\ (|0> - |1>) = (\cos(\pi/4) - \sin(\pi/4))'$. Finally, $q_y = (0\ 1\ 0)$ locates the qubit $|q_y> = 1/2^{1/2}\ (|0> + i|1>) = (\cos(\pi/4)$ $i\sin(\pi/4))'$. Notice that $R_Z(\pi)$ applied to $|q_x>$ is $(-i\cos(\pi/4)\ i\sin(\pi/4))' = -i|q_{-x}>$, where the global phase $-i$ is irrelevant because single qubits are in a projective space CP^1. Next, $R_Z(\pi/2)$ multiplying $|q_x>$ gives $1/2\ (1-i\ \ 1+i)' = 1/2\ ((1-i)\ \cos(\pi/4)\ (1+i)\ \sin(\pi/4))' = (1-i)/2^{1/2}\ (\cos(\pi/4)\ (1+i)/(1-i)\ \sin(\pi/4))' = |q_y>$. Again, the global phase does not matter. Lastly, using $R_Y(\pi/2)$, multiplying $|q_x>$ yields the south pole qubit, $|1>$. The matrix representation of $2^{1/2}\ R_Y(\pi/2) = \begin{matrix} |1 & -1| \\ |1 & 1|. \end{matrix}$

Multiplying $R_Y(\pi/2)$ times the column vector, $|q_x> = 1/2^{1/2}\ (1\ 1)'$, yields the qubit representing the south pole, $|1>$.#

The Pauli rotational operators have a sort of universal property in that for any U in U(2, C), there are real numbers a, b, c, and d such that the unitary operator can be expressed in terms of three Pauli operators: $U = e^{ia}\ R_Z(b)\ R_Y(c)\ R_Z(d)$ (Nielsen and Chuang, 2000).

Example 16.11:

Notice that the Hadamard gate H can be represented using the aforementioned expression, namely by $H = e^{ia}\ R_Z(0)\ R_Y(\pi/2)\ R_Z(\pi)$. Here, $R_Z(0) = I$, and given below are $R_Y(\pi/2), R_Z(\pi)$ and $R_Y(\pi/2)R_Z(\pi)$ all in order:

$$\begin{matrix} |c\ \pi/4 & -s\ \pi/4 & |-i & 0| & |-i/2^{1/2} & -i/2^{1/2}| \\ |s\ \pi/4 & c\ \pi/4| & |0 & i| & |-i/2^{1/2} & -i/2^{1/2}|. \end{matrix}$$

By letting $e^{ia} = i$, that is, $a = \pi/2$, the last rightmost matrix above is H. Other gates in the previous section are even easier to represent with the Pauli rotation operators. For instance, the shift gate $S = e^{i\pi/4}\ R_Z(\pi/2)$.#

The Clifford single-qubit group is generated by the Hadamard and phase-shift matrices H and S. This group is defined as the set of unitary matrices U, which when applied to a Pauli matrix P, as UPU* this quantity yields a Pauli matrix again. The cardinality of this group is 24.

Example 16.12:

Note that SXS* = Y, SZS* = Z, HXH* = Z, and HZH* = X. For the last identity, note that the quantity $2^{1/2}$ HZ and HZH* are given in order and are equal to:

$$\begin{matrix} |1 & -1| & |0 & 1| \\ |1 & 1| & |1 & 0|.\# \end{matrix}$$

16.4 Multiple-qubit input gates

As previously mentioned, operations involving multiple qubits are very often inputted with these qubits tensored together. For the case involving two qubits, the operation X: H ⊗ H→H ⊗ H. As an instance of this is the operator X = CNOT. Most often, for a multiple-qubit gate, the first qubit is used as a control qubit. This means that the actual operation is employed to the other, the second qubit. Here, the second qubit is conditioned on the state of the first qubit, and the second qubit is called the target. An example of this type of operation is detailed next.

Consider the CNOT gate, also known as the conditional not operation. The input to this operation is $|x\rangle \otimes |y\rangle$, where ⊗ denotes the tensor operation. The tensored product is written more briefly as $|x\ y\rangle$, where x is the first or top qubit and the second is y. Both x and y can take on only the values of zero or one. This gate is illustrated in Fig. 16.2.

As can be seen from this diagram, the first qubit $|x\rangle$ passes through not altered. The bottom output is the qubit $|y\ (+)\ x\rangle$, where (+) denotes the mod2 operation or exclusive or. Thus $|y\ (+)\ x\rangle = 0$ when and only when x = y; otherwise it equals 1. The CNOT operation is given in C^4 by the matrix:

$$\begin{matrix} |1 & 0 & 0 & 0| \\ |0 & 1 & 0 & 0| \\ |0 & 0 & 0 & 1| \\ |0 & 0 & 1 & 0|. \end{matrix}$$

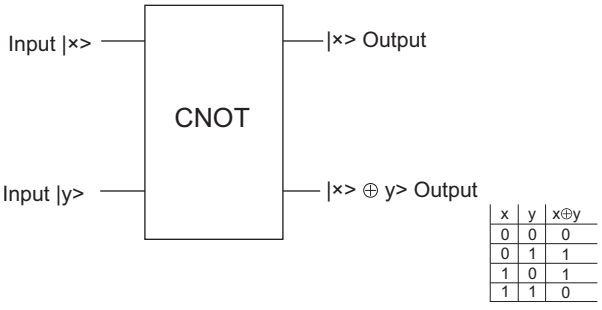


FIGURE 16.2 CNOT gate.

The aforementioned CNOT matrix is symmetric and an involution; this shows that the CNOT operation is unitary. This matrix multiplies $|x\ y\rangle$ when expressed as a column vector in C^4 after the tensor product is utilized. There are four possibilities for this input; they are as follows: $|0\ 0\rangle = |0\rangle \otimes |0\rangle = (1\ 0)' \otimes (1\ 0)'$, $|0\ 1\rangle = (1\ 0)' \otimes (0\ 1)'$, $|10\rangle = (0\ 1)' \otimes (1\ 0)'$, and finally, $|11\rangle = (0\ 1)' \otimes (0\ 1)'$. The 4x1 vector representations are given as column vectors in the corresponding order:

$$\begin{matrix} |1| & |0| & |0| & |0| \\ |0| & |1| & |0| & |0| \\ |0| & |0| & |1| & |0| \\ |0| & |0| & |0| & |1|. \end{matrix}$$

The matrix CNOT when operating on these vectors in the exact same order is as follows:

$$\begin{matrix} |1| & |0| & |0| & |0| \\ |0| & |1| & |0| & |0| \\ |0| & |0| & |0| & |1| \\ |0| & |0| & |1| & |0| \end{matrix}$$

Now these resulting vectors must be interpreted in terms of tensored kets.

Representing the 4x1 vectors above, in order and in terms of kets, gives $|0\ 0\rangle, |0\ 1\rangle, |1\ 1\rangle$, and finally, $|1\ 0\rangle$. Accordingly, when the first qubit is zero, the output is $|x\ y\rangle$ and is unchanged. When the first qubit is one, then the second qubit flips sign, that is, the output becomes $|x\ (1-y)\rangle$. The determinant of CNOT is minus one, so it is in the Lie group U(4, C), but it isn't in the Lie group SU(4, C).

Notice that the CNOT operator is an entangler, but also it is an unentangler. For instance, $\text{CNOT}(1/2^{1/2}\ (|0\ 0\rangle + |1\ 0\rangle)) = 1/2^{1/2}\ (|0\ 0\rangle + |1\ 1\rangle)$; the last qubit pair is entangled. It is one of the Bell states. To see this, first note that $|+0\rangle = 1/2^{1/2}\ (|0\rangle + |1\rangle) \otimes |0\rangle$ so the input to the CNOT is not entangled. Finally, to see that the output is entangled, note that the most general $|0\rangle$ and $|1\rangle$ qubit tensor product is $z = a11|0\rangle \otimes |0\rangle + a12|0\rangle \otimes |1\rangle + a21|1\rangle \otimes |0\rangle + a22|1\rangle \otimes |1\rangle$. The tensor is called a pure tensor iff a11 a22 = a12 a21; otherwise it is entangled. Here a21 = 0, but a11 = a22 = 1. See Section 7.4 for further information on entanglement.

Finally, using the Pauli matrices X, Y, and Z are as follows:

$$\begin{matrix} |0 & i| & |0 & -1| & |i & 0| \\ |i & 0| & |1 & 0| & |0 & -i| \end{matrix}$$

$$\text{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = (I + Z) \otimes I + (I - Z) \otimes X/2$$

In higher dimensional Hilbert spaces, unitary operators can be replaced by the CNOT gate in U(4,C) along with single-qubit gates in U(2,C) (Nielsen and Chuang, 2000)

The Pauli group on n qubits consists of $\{1, -1, i, -i\} \times \{I, X, Y, Z\}^{@n}$, where the last set involves the Pauli matrices tensored together n times. In a similar manner as in the single-qubit case, the Clifford n qubit group consists of all the unitary matrices U such that for P and P′ in the n qubit Pauli group, $U\ P\ U^* = P'$.

Example 16.13:

For the two qubit case, notice that when using operations in order by multiplying the following three matrices, CNOT, (Z ⊗ I), CNOT*, the result is the Z ⊗ I. That is, CNOT.(Z ⊗ I). CNOT* = Z ⊗ I. Likewise, CNOT.(I ⊗ Z). CNOT* = Z ⊗ Z, CNOT.(I ⊗ X). CNOT* = I ⊗ X, and CNOT.(X ⊗ I). CNOT* = X ⊗ X. For the last identity, note that X ⊗ I, CNOT.(X ⊗ I), and X ⊗ X are given in order as follows:

$$\begin{array}{cccc|cccc|cccc|}
|0 & 0 & 1 & 0| & |0 & 0 & 1 & 0| & |0 & 0 & 0 & 1| \\
|0 & 0 & 0 & 1| & |0 & 0 & 0 & 1| & |0 & 0 & 1 & 0| \\
|1 & 0 & 0 & 0| & |0 & 1 & 0 & 0| & |0 & 1 & 0 & 0| \\
|0 & 1 & 0 & 0| & |1 & 0 & 0 & 0| & |1 & 0 & 0 & 0|.
\end{array}$$

16.5 The swapping operation

The CNOT gate is additionally useful in creating other important gates. In particular, the swap gate is designed by utilizing three CNOT gates in succession. See Fig. 16.3. The symbol for the swap gate is SWAP. When used SWAP: $H_A \otimes H_B \rightarrow H_B \otimes H_A$. For input state $|x\ y\rangle$, the swap gate yields SWAP($|x\ y\rangle$) = $|y\ x\rangle$. It begins with the first bit as a control bit in the first CNOT. Then the second bit is used as a control bit in the second CNOT, and finally for the last CNOT, the first bit is used again as the control bit. The following sequence illustrates the swapping operation, with a single arrow indicating the use of a CNOT operation: $|x\ y\rangle \rightarrow |x, x\ (+)\ y\rangle \rightarrow |x\ (+)\ x\ (+)\ y, x\ (+)\ y\rangle = |y, x\ (+)\ y\rangle \rightarrow |y, x\ (+)\ y\ (+)\ y\rangle = |y\ x\rangle$.

16.6 Universal quantum gate set

Intuitively a universal quantum gate set (UQGS) is a fixed, finite number of unitary operators that, when applied in succession, can approximate any unitary operation in SU(2, C). The operators are applied in composition, from left to right. However, they are illustrated as the concatenation of strings of operations, labeled in order. Let $G = \{g_1, g_2, \ldots, g_m\}$ be a finite subset of SU (2, C). A word of length n is $w_n = g_1\ g_2 \ldots, g_n$, with g_i in G and n greater or equal to 1. The set of all words from G with length n or less is denoted by G_n. The union of all G_n for $n < \infty$ is

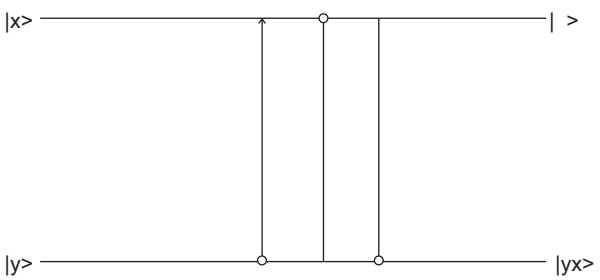


FIGURE 16.3 The SWAP Gate.

denoted by <G>. A set of gates is UQGS whenever <G> is dense in SU(2, C). That is, for any U in SU(2, C), and $\epsilon > 0$, there exists a string of operators g in <G> such that the distance d (U, g) is less than ϵ. The metric is induced by the trace norm, $\|A\| = \text{tr}[(A^* A)^{1/2}]$. The trace norm satisfies the equational identities and inequalities:

1) Triangle Inequality: $\|A+B\|$ is less than or equal to $\|A\|+\|B\|$.
2) Submultiplicative: $\|A\ B\|$ is less than or equal to $\|A\|\ \|B\|$.
3) Unitary Invariance: $\|U\ A\ V\| = \|A\|$ for all unitary operators U and V.

The operator norm is also used; it is $\|A\| = \sup |Av|$, for $\|v\| = 1$. For U and V in U(2,C), U approximates V that means d (U, V) = sup $\|(U-V)\ v\| < \epsilon$, for $\|v\| = 1$. For two strings of quantum gates $U = U_1\ U_2 \ldots, U_n$ and $V = V_1\ V_2 \ldots, V_n$, then d(U,V) is less than or equal to the sum, Σ d (U_i, V_i). This follows using n = 2 and can be proved by mathematical induction. For n = 2, $d(U_1\ U_2, V_1\ V_2) = \sup \|(U_2U_1-V_2V_1)\ v\| = \sup \|(U_2U_1-V_2U_1+V_2U_1-V_2V_1)\ v\|$ is less than or equal to $\sup \|(U_2U_1-V_2U_1)\ v\|+\sup\|(V_2U_1-V_2V_1)\ v\ \| = \sup\|(U_2-V_2)\ U_1v\ \|+\sup \|V_2\ (U_1 - V_1)\ v\| = d(U_2, V_2)+d(U_1, V_1)$. Note that a maximum could be used here in place of sup and that the order of operation of operators is backward from the string notation.

A most popular UQGS for SU(2, C) consists of the Hadamard gate H and the $\pi/8$ gate T. The proof makes use of the Pauli rotation operators. Section 16.3 illustrates that the Hadamard gate is a product of Pauli rotation operators. Also, $T = R_Z(\pi/4)$. The T gate and the HTH string of gates are successively utilized in showing {H, T} is a UQGS. An important identity in producing the approximation with any degree of accuracy is the identity $THTH = R_Z(\pi/4)R_X(\pi/4)$. More general UQGS is explained in Lloyd (1995).

16.7 The Haar measure

Haar measure u is best understood as a left G invariant measure (Nachbin, 1965). Here G is a locally compact topological group. For the binary operation of multiplication, an invariant measure is such that u(x V) = u (V), where x is an element of the group G and V is a Borel measurable set. Appendix A.2 provides an outline of relevant measure theoretic concepts.

Historically, Haar measure was first expressed as an approximation of an absolute measure based on a unit measure. The unit measure for the Haar measure uses E as a Borel subset of G, and V is a nonempty open subset of G. Then let (E: V) denote the smallest number of left translates of V that cover E. Thus, (E: V) = inf {cardinality of A such that E is a subset of the union of x V, for x in A, A a subset of G}. Hence (E: V) can be viewed as a relative measure of E.

Fix a precompact open subset Eo, and assign its measure as one. Precompact means that the closure of the set is compact. Then consider as an approximation to the absolute measure of E the following: (E: V)/(Eo: V). Notice for all x in G, (E: V) /(Eo:V) = (Ex: V)/(Eo: V), and this operation is left G invariant. Taking the limit as V shrinks to the identity element shows that the Haar measure is defined as the limit. So, u(E) = limit [(E: V)/(Eo: V)], as $V \to I$.

For G, a locally compact topological group, Haar's theorem states that there is up to a positive multiplicative scalar a countably additive measure on Borel subsets of G such that

1) The measure is left invariant as mentioned earlier.
2) For a compact subset K of G, $u(K) < \infty$.

3) The measure is both outer regular on Borel sets in G and inner regular on open sets U in G, respectively, u(S) = inf {u(U) such that S is a subset of an open set, U in G}, and u (U) = sup {u(K) such that K is a subset of U and K is compact}.

Example 16.14:

Consider the carrier set for G to be the group of positive real numbers R + , under multiplication, with the usual topology on the real line. Then for any Borel subset, B, in R, a Haar measure can be created using the usual Cauchy-Riemann, Riemann, or Lebesgue integral: $u(B) = \int_B 1/x\ dx$. For any Borel set, say the interval [2, 4], the corresponding measure u([2, 4]) = ln(4)−ln(2) = ln(2). Also for any positive real number in R + , say three, and multiplying the interval by three, then the Haar measure u([6, 12]) = ln(12)−ln(6) = ln(2) again. This illustrates the invariance of the Haar measure using the multiplicative group involving elements in R + .#

16.8 Solovay–Kitaev theorem

The notion of unitary operators in SU(2, C) being approximated by a string of unitary operators, $S_1S_2...S_n$, is described in Section 16.6. This string forms a UQGS and was explained with emphasis on {H, T}, the Hadamard, and the $\pi/8$ gate set. In this section, the degree of closeness in the approximation is related to the number of elements n, in the string of unitary operators.

The Solovay–Kitaev theorem (SK) gives a constructive procedure to approximate arbitrary unitary 2x2 matrices with determinant one by a string of physically realizable unitary gates. The approximation is to within $\epsilon > 0$. The string is a finite product of gates from a universal gate set S. The SK result shows that the length of the string of gate products needed is of the order $O(\log^4(1/\epsilon))$. Required, for these approximations to hold, the set of gates S must be dense in G. More precisely, consider a finite subset S, of SU(2, C), which contains inverses and is dense in SU(2, C). Then there is a constant c, in [1, 4], such that for any such S and $\epsilon > 0$, n can be chosen where $n = 0(\log^c(1/\epsilon))$ such that S^n is an ϵ net for SU(2, C). For any U in SU(2, C), there is a string of gates $S_1S_2...S_n$ in S^n, where $d(S_1S_2...S_n, U) < \epsilon$. The distance metric d is induced by the trace norm. That is, d (A, B) = $\|A-B\|$. So for A, B in SU(2, C), $\|A\| = \text{Tr}\,[(A\ A^*)^{1/2}]$, this is also equal to the sum of the absolute value of the two eigenvalues of A. In short, an ϵ net for SU(2, C) means that SU(2, C) is a finite subset of the union of open balls of radius ϵ centered about strings $S_1S_2...S_n$ in S^n. Thus, for every unitary operator in SU(2, C), there exists a string of gates from S^n that come ϵ close to producing the same results.

The SK criteria for closeness in approximation are inversely related to an increase in the length of the approximating string of operators. The group commutator $[U, V]_G = UVU^*V^*$ is employed in the proof of SK. It is used in determining how close unitary operators are to the identity operator, I. The group commutator is related to the usual Lie bracket for U, V in SU(2, C), as follows: [U, V] = 0, then $[U, V]_G = I$. This is true because [U,V] = 0, which means that UV = VU; then in the group commutator, $[U, V]_G = UVU^*V^*$, replacing UV by VU gives $[U, V]_G = VUU^*V^*$, which is the identity.

Consequently, the group commutator is close to the identity when and only when the two operators commute. Repeated application of this commutator relationship enables ϵ net approximations to be made to the identity element. Lastly by translation, the approximation can be seen to hold for any unitary operator in SU(2, C). Most proofs of the SK theorem employ the shrinking lemma specifying better approximations of the identity using larger length strings. Smaller and smaller ϵ net approximations are made to neighborhoods of the identity. The final step in the proof is using the bi-invariance of distance to translate the ϵ nets to approximate other elements in SU(2, C). In-depth details are provided in Zarapico (2018). For S in a UQGS with algebraic entries and $\epsilon > 0$, n can be chosen where $n = 0(\log^c (1/\epsilon))$, with $c = 1$ (Holevo, 2003; Bourgian and Gamburd, 2001).

16.9 Quantum Fourier transform and phase estimation

The quantum Fourier transform (QFT), denoted by F, sends vectors in a complex Hilbert space C^N back into C^N, where usually N is the positive integer 2^n. Specifically, for $|x> = \sum_{j=0}^{N-1} x_j|j>$, the QFT, $F(|x>) = \sum_{j=0}^{N-1} y_j|j>$, where $y_j = 1/N^{1/2} \sum_{k=0}^{N-1} x_k(w_N)^{km}$, for $m = 0, 1, 2, \ldots, N-1$. Also, $w_N = e^{(2i\pi)/N}$. The inverse QFT is similarly defined. It too sends vectors in the complex Hilbert space C^N back into C^N. In this case, $x_j = 1/N^{1/2} \sum_{k=0}^{N-1} y_k(w_N)^{-km}$, for $m = 0, 1, 2, \ldots, N-1$. The QFT as well as its inverse can be implemented as a unitary matrix, U_N operating on quantum state vectors.

Example 16.15:

In the Hilbert space C^2, the QFT, $F = H$, the Hadamard transform. This follows since $w_N = e^{(2i\pi)/N} = w_2 = e^{(2i\pi)/2} = -1$. Therefore, $|x> = \sum_{j=0}^{1} x_j|j>, F(|x>) = \sum_{j=0}^{1} y_j|j>$, where $y_j = 1/2^{1/2} \sum_{k=0}^{1} x_k(w_2)^{km}$, for $m = 0, 1$. $F(|x>) = 1/2^{1/2}[x_0(w_2)^0 + x_1(w_2)^0]|0> + 1/2^{1/2}[x_0(w_2)^0 + x_1(w_2)^1]|1> = 1/2^{1/2}[x_0 + x_1]|0> + 1/2^{1/2}[x_0 - x_1]|1>$. So it follows that $F(|x>) = H\ (x_0\ x_1)'$.#

Example 16.16:

In C^4, let $|x> = \sum_{j=0}^{N-1} x_j|j> = 1/3(|0> + 2i|2> - 2|3>) = 1/3(|00> + 2i|10> - 2|11>)$. Notice that the norm $\| |x> \| = 1$. The objective is to find the QFT, $F(|x>)$. This will be computed using the unitary matrix U_4. Provided below is $2U_4$, followed by the vector, $3|x>$ represented as a four-by-one column vector in C^4,

$$\begin{matrix} |1 & 1 & 1 & 1| \\ |1 & i & -1 & -i| \\ |1 & -1 & 1 & -1| \\ |1 & -i & -1 & i| \end{matrix} \quad \begin{matrix} | & 1 & | \\ | & 0 & | \\ | & 2i & | \\ | & -2 & |. \end{matrix}$$

Multiplying out the four-by-four matrix $2U_4$, time the four-by-one vector, $3|x>$, in C^4, gives $F(|x>) = 1/6\ (-1+2i \quad 1 \quad 3+2i \quad 1-4i)'$. Note that the norm of $F(|x>) = 1$. This had to be the case since a unitary mapping is always an isometry; thus it preserves the length of vectors.#

A quantum circuit to perform the QFT involves inputting each qubit and then using first a Hadamard matrix followed by a sequence of several controlled rotation matrices of the form:

$$\begin{vmatrix} 1 & 0 \\ 0 & e^{2i\pi/(2^k)x_j} \end{vmatrix}$$

This is employed for each input qubit.

16.10 Uniform superposition and amplitude amplification

In quantum algorithms, an important application of the Hadamard gate is to utilize it on almost every input. Usually, the inputs are n in number and are all $|0>$ qubits. In this case, the operation is called the diffusion operation. Say that a Hadamard operator is applied to every input in parallel, it is symbolized as $H^{@n}\left(|0>^{@n}\right) = 1/2^{n/2}\sum_{x=0}^{N-1}|x>$, where $N=2^n$, and the tensor product is symbolized by @. So the probability associated with any vector $|x>$ within the uniform superposition is $1/2^n$. Since all amplitudes are the same, the average height of any $|x>$ is also $1/2^n$.

Example 16.17:

Consider the two input system, $H^{@2}\left(|0>^{@2}\right) = 1/2\sum_{x=0}^{3}|x>$. Mathematically, this follows because $H|0> = 1/2^{1/2}(|0> + |1>) = |+>$, and since there are two Hadamard gates in parallel, the computation is $H^{@2}(|0>^{@2})$. So forming the tensor product of $|+>$ with itself, that is, gives $|+> @ |+> = 1/2\ [|0\ 0> + |0\ 1> + |1\ 0> + |1\ 1>]$. Interpreting each ket, $|x>$ in binary, provides the same result, namely $1/2\sum_{x=0}^{3}|x>$. A simple calculation shows that these vectors within the uniform superposition are each of unit length and mutually orthogonal to each other. The last property is best seen using the representation in C^4. In this case, each of these kets is represented by a four-by-one vector. Each vector contains all zero-valued tuples except for a single one in different locations.#

Letting $|w>$ be the uniform superposition of $N=2^n$ states $|w> = 1/2^{n/2}\sum_{x=0}^{N-1}|x>$. Then these vectors are often illustrated in a bar-type graph with N vertical line segment entries on the abscissa all of height $1/2^{n/2}$. See Fig. 16.4A. Uniform superposition is usually the initial state for an amplitude amplification algorithm. As a prime example is the Grover search, described in the next chapter. In its simplest form, this search utilizes the pair $\{R_W, U_f\}$, consisting of two reflection operators to find a single vector $|x^*>$ within $|w>$. An application of the first reflection U_f negates the value of $|x^*>$ or reflects it about the abscissa. That is, it becomes $-|x^*>$, or graphically it reflects the state $|x^*>$ in the line segment graph; see Fig. 16.4B. This operator is also called an oracle. In this diagram, a dashed horizontal line indicates the new lower average value u, of all probability amplitudes for states in $U_f|w>$. The overall lower average or the overall lower mean value of the probability amplitudes occurs because the application of U_f kept all states invariant but it

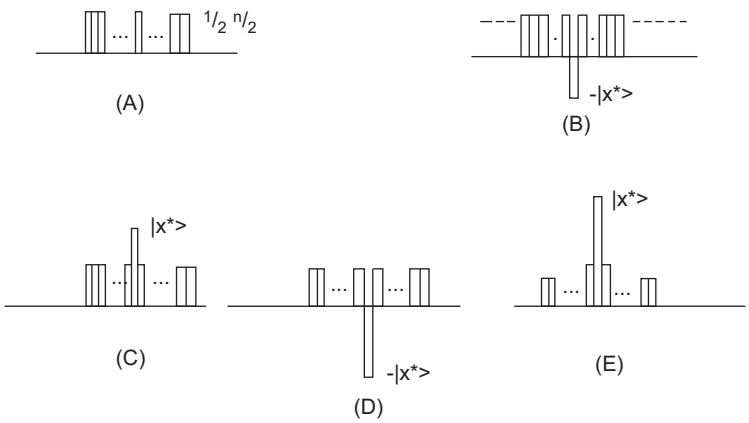


FIGURE 16.4 Trace of amplitude amplification. (A) Uniform superposition, (B) Reflection of the state $|x^*>$, (C) Amplitude amplification, (D) One and a half reflection pairs applied again, (E) Two refection pairs applied/.

negated $|x^*>$. So this single negation pulls the overall average amplitude down slightly. The average has nothing to do with the true average involving the actual probability. This follows since the actual probabilities of $|x^*>$ and $-|x^*>$ are the same. The absolute amplitude squared gives the true probability, and therefore, this quantity does not change.

The next reflection operation R_W is defined herein to be a reflection of the mean or actually, the average value u of the amplitude. Accordingly, for any vector $a_y|y>$ with $|y>$ in $\{0, 1\}^n$, the reflection operation about this mean u is such that $R_W(a_y|y>) = (2u - a_y)|y>$. This equation is shown to hold by subtracting u from height a_y, obtaining $(a_y - u)$. Then reflect this quantity about the x-axis as in an application of U_f. This operation only negates the quantity. Accordingly, the result is $(u - a_y)$, and finally lifting, that is adding u again provides the desired result.

Since the mean is smaller for all vectors $U_f|x>$ in $U_f|w>$, except for $U_f|x^*>$, applying the operator R_W will make these amplitudes smaller yet. However, the amplitude of $R_W U_f|x^*>$ will become about three times its original value. See Fig. 16.4C, which is not to scale. Amplitude amplification is the name given to the process of utilizing a pair of reflections such as R_W and U_f in succession. If the pair of reflections were applied again, then first, $U_f R_W U_f|w>$ is illustrated in Fig. 16.4D, and $R_W U_f R_W U_f|w>$ is illustrated in Fig. 16.4E. In any case, repetitive applications of the reflection pair produce a decreasing average value of all state amplitudes while increasing the probability amplitude of the desired state. This beneficial increase occurs provided that $\pi/4N^{1/2}$ applications of $R_W U_f$ are not exceeded.

These reflections are the corner stone for the Grover search. This search is useful for logistic applications like finding the most efficient path and similar optimization problems. It is also used in machine learning algorithms, in particular, classification problems. Also, one-way functions that are hard to calculate, but easy to verify provide good results with the Grover search method.

16.11 Reflections

Reflections in the finite Hilbert space $H = C^2$ have a precise meaning. Reflections are used as a linear operator R_W, R_W: $H \to H$. The operator is always applied to a nonzero

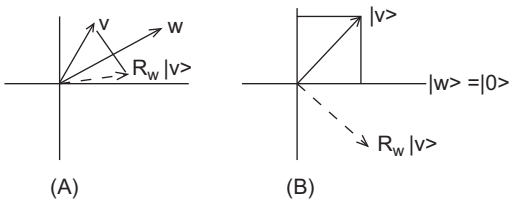


FIGURE 16.5 Reflection operation.: (A) Two vectors and reflection ket, (B) Result of reflection operation.

vector $|v>$. By doing so, it reflects this vector $|v>$ about a specified nonzero vector $|w>$. See Fig. 16.5A, where $|v>$ and $|w>$ are illustrated along with the reflected ket, $R_W|v>$. The end result is written with a dashed line. A most simple instance of this is when $|w> = |0> = (1\ 0)'$ in the computational basis, and $|v> = (a\ b)'$. Then $R_W|v> = (2|0><0|-I)|v> = (a\ -b)'$. This operation is also called a Householder reflection (Householder, 1958). As in the last section, use the diffusion operator to represent $|w> = H^{@n}\ (|0>^{@n})$. Then $R_W|v> = H^{@n}\ [2|0^{@n}>< 0^{@n}|-I]\ H^{@n}|v> = (2|w><w|-I)|v>$.

Accordingly, the operator performing the reflection about w is given by $R_W = (2|w><w|-I)$, where I is the identity operation.

Example 16.18:

Let $|w> = |0>$ be the simple vector about which a reflection is performed. If the vector to be reflected is $|v> = (3/5)^{1/2}|0> + (4/5)^{1/2}|1> = ((3/5)^{1/2}(4/5)^{1/2})'$ in C^2, the operator to perform the reflection involves twice the pure density function $|w>< w|$, along with the negative of the identity function. Thus $R_W =$

$$\begin{matrix} |1 & 0| \\ |0 & -1|. \end{matrix}$$

An application of this operator upon the vector $|v>$ gives $R_W|v> = (3/5)^{1/2}|0> - (4/5)^{1/2}|1> = ((3/5)^{1/2} - (4/5)^{1/2}))'$ in C^2. See Fig. 16.5B.#

The reflection operator is a key operation in the Grover search methodology. This operator is performed over and over again along with a sign alternating operation to perform this search.

References

Bourgian, J., Gamburd, 2001. A special gap theorem in SU(d). arXiv: 1108.6264.

Bennett, C., Bernstein, E., Brassard, G., Vazirani, U., 1997. Strengths and weaknesses of quantum computing. SIAM J. Comput. 26 (5), 1510–1523.

Holevo, A., 2003. Classical capacities of quantum channels with constrained inputs. Probab. Theory Appl. 48, 359 (quant–ph)/(0211179).

Householder, A., 1958. Unitary triangularization of a nonsymmetric matrix. J. ACM .

Lloyd, S., 1995. Almost any quantum logic gate is universal. Phys. Rev. Lett. 75.

Nachbin, L., 1965. The Haar Integral. Van Nostrand.

Nielsen, M.A., Chuang, I.L., 2000. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

Zarapico, J., 2018. Efficient Unitary Approximations in Quantum Computing: The Solovay-Kitaev Theorem (thesis). University of Barcelona.

This page intentionally left blank

CHAPTER

17

Quantum computing applications

17.1 Deutsch problem description

The Deutsch problem is one of the earliest problems having a quantum solution outperforming a classical solution. The problem involves a function f, f: $\{0, 1\} \rightarrow \{0, 1\}$, and the objective is to determine whether the function is constant or if it is balanced. To be constant means that the output is always the same, no matter what the value of the input. To be balanced means that the output has a zero for one value of the input and a one for the other value of the input. The five columns below illustrate all the possibilities. The first column provides the input to the function f; it has as tuples a zero or a one. The next two columns give the outputs from f, illustrating the constant concept. The last two columns are the outputs of a balanced function f.

|0|	|0|		|1|	|0|		|1|
|1|	|0|		|1|	|1|		|0|
INPUT	^	CONSTANT	^	^	BALANCED	^

Classically to determine whether f is constant or balanced, it is obvious that two queries are needed. First f must be evaluated for the input value, say zero, and next it must be evaluated at the other input value, one. This is indicated above with two possibilities for a constant function and two possibilities for a balanced function. A quantum solution could determine if f is constant or balanced using only one quarry. Details involve the use of an oracle, which is a unitary transformation U, represented as a black box illustrated in Fig. 17.1.

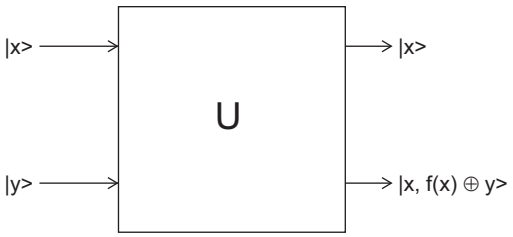


FIGURE 17.1 Deutsch oracle.

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00002-3

17.2 Oracle for Deutsch problem solution

In Fig. 17.1, the oracle is given as a black box; the only thing that is known is that U maps four dimensions into itself. U: C^2 (x) $C^2 \rightarrow C^2$ (x) C^2, where U is an operation such that U (|x> |y>) = |x>|(y(+) f(x))>, for all x, y, f(x) in {0, 1}. The addition (+) is the mod(2) operation, the same as exclusive or, in logic. So whenever y and f(x) are of the same value, zero is the result; otherwise one is the result. Only unitary operations are allowed. Proving that U is a unitary operator follows by showing that U is an involution, that is, $U^2 = I$, the identity function. Also required is that U is a normal operator, that is, U*U = UU*. First, an involution is shown by applying U twice in succession, so U^2(|x>,|y>) = |x> |(y(+) f(x) (+) f(x))> = |x> |y>; this shows an involution because f(x) (+) f(x) = 0. The normality of U will be immediate once the black box or oracle is identified as the symmetric matrix U =

$$\begin{vmatrix} 1-f(0) & f(0) & 0 & 0 \\ f(0) & 1-f(0) & 0 & 0 \\ 0 & 0 & 1-f(1) & f(1) \\ 0 & 0 & f(1) & 1-f(1) \end{vmatrix}$$

Multiplying the second input qubit |y>, represented in C^4, by U always provides the second output of the oracle. Illustrations are given later.

The matrix U incorporates all four situations involving all the possibilities for a pair of input qubits, |x> the control and |y> the target, along with the output pair of qubits, |x> and |y (+) f(x)>, where x, y, and f(x) are 0 or 1, and (+) is mod 2. These four different scenarios are listed below in the last four columns. The first column denotes the input qubit pair of values for |x y>. The last four columns give the output value of |x y(+) f(x) using the four cases: 1) f(0) = f(1) = 0; 2) f(0) = 0, f(1) = 1; 3) f(0) = 1, f(1) = 0; 4) f(0) = f(1) = 1. These are given in order below where situations 1) and 4) correspond to constant f, and 2) and 3) are for balanced f. Inspection of the second, that is, the right-hand, column for all situations provides the results.

	1)	2)		3)	4)
00	00	00		01	01
01	01	01		00	00
10	10	11		10	11
11	11	10		11	10
INPUT	CONSTANT	^	BALANCED	^	CONSTANT

There are four matrices associated with U, and all are given consisting of 2 x 2 block submatrices involving the identity matrix or the Pauli x, or Pauli s1, matrix. For instance, for case 3), the matrix U has the Pauli x matrix in the upper left-hand corner, and it has the identity two by two in the lower right corner, U =

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

As previously mentioned, when this matrix multiplies the input vector in C^4 representing input qubit |y>, that is, (0 1 0 1)′, the result is (1 0 0 1)′, which is the second output in situation 3).

Fig. 17.2 provides a quantum circuit for each of the four situations mentioned earlier. In circuits for cases 2) and 3), there are CNOT gates. In cases 3) and 4), there are single-qubit NOT gates.

17.3 Quantum solution to Deutsch problem

Numerous quantum solutions exist for solving the Deutsch problem; however, they all involve the Deutsch oracle. Perhaps the easiest solution is illustrated in Fig. 17.3. Here, the actual inputs to the circuit illustrated in this figure are both the same qubit |1>. Using the tensor product yields as input |1> @ |1> = |11>. Referring to this diagram, after the first Hadamard operation on both the first and second qubit yields the state value of |v1> = 1/2 (|00> − |10> − |01> + |11>), this is a superposition of all possible

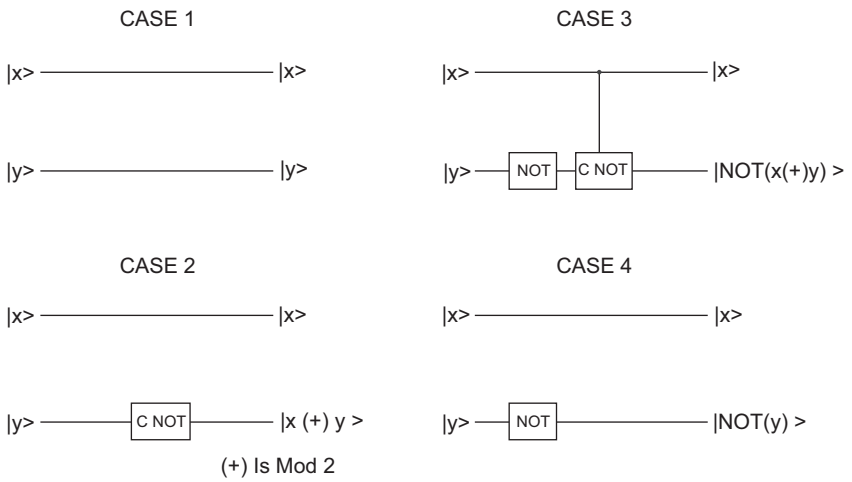


FIGURE 17.2 Quantum circuits representing Deutsch oracle.

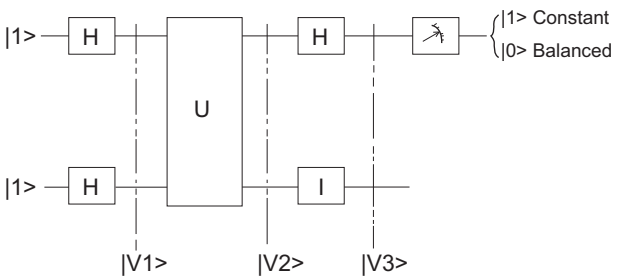


FIGURE 17.3 Deutsch algorithm.

combinations of the inputs, but is not entangled. The computation in determining |v1> is the tensor product involving the Hadamard transform: (H|1>)@(H|1>) = 1/2 (|0> − |1>)@ (|0>− |1>) = 1/2 (|00>− |10>− |01>+ |11>).

The next state value, after the oracle U, yields the quantity: |v2> = 1/2(|0 f(0)>− |1 f(1)>)− |0 1(+)f(0)>+ |1 1(+)f(1)>. Now if f is constant, that is, f(0) = f(1), then |v2> = 1/2 (|0>− |1>)(|f(0)>− |1(+)f(0)>). On the other hand, if f is balanced, then f(0) = 1(+)f(1); in this case, |v2> = 1/2(|0>+ |1>)(|f(0)>− |1(+)f(0)>). Finally, utilizing the Hadamard on the first qubit and the identity on the second qubit yields the results that |v3> = |1> (|f(0)>− |1(+)f(0)>) occurs whenever f is constant. Also, |v3> = |0> (|f(0)>− |1(+)f(0)>), whenever f is balanced. Next measuring the contents of the first register yields |1> when f is constant. Additionally it gives an observed value of |0> for f balanced. Only one cycle execution was needed to obtain this result.

17.4 Deutsch-Jozsa problem description

This problem is among the earliest to have a quantum solution outperforming a classical one. The problem involves a function f, f: $\{0, 1\}^n \rightarrow \{0, 1\}$, n > 1, and the objective is to determine whether the function is constant or if it is balanced. Since the output is a single value, to be constant means that the output is always the same, no matter what the value of the input. To be balanced means that the output has zeros for half of the inputs and ones for the other half of the inputs. Additionally, it is promised that these are the only types of outputs; no others exist. The problem is to determine for sure if the function f is constant or balanced. The quantum solution only utilizes a single evaluation.

To analyze the situation, it is assumed that the input is utilized 2^n times in succession to obtain an output string consisting of 2^n zeros or ones. For the 2^n different possible input values, there are only two constant strings of outputs; these are all-zero or all-one strings. Correspondingly, there are 2^{2^n} possible strings of a mixed number of zeros or ones. Only some of these strings contain an equal number of zeros and ones. For the case in which f is balanced, half the length of a string must contain zeros and the other half must contain ones, not in any order. Since the length of the output string is 2^n, a permutation consisting of this many distinct objects can be one of $(2^n)!$ That is, the only possible number of outputs is 2^n factorial. However, for the balanced case, half or 2^{n-1} values of an output string must be zeros and the other half must be ones. Accordingly, there are $(2^n)!/[(2^{n-1})!\ (2^{n-1})!]$ balanced output strings. For instance, if n = 3, then, again there are only two constant output strings, but there are 8!/[4! 4!] = 70 balanced output strings. In any case, if all output strings are equally likely to occur, then for n = 3, the probability of obtaining a constant string would be 1/35. As n gets larger, this probability goes to zero, but the result is not certain. To be certain in classical methods, the first 2^{n-1} output strings must be examined. After this, and when and if they are all of the same value, then the next, $2^{n-1} + 1$ string will determine for sure if f is constant or balanced.

Example 17.1:

Using n = 2, the Deutsch-Jozsa problem will be illustrated. In this case, the input consists of two tensored qubits. But the output corresponding to this pair of input qubits is a

single qubit, as in the Deutsch problem. Below in the left column are all the possible input pairs of qubits. These qubits are denoted without the ket notation, and output strings are presented as columns. The next two columns show the only possible constant output values; these are output strings of length four of all zeros and all ones, respectively. The final six columns show the balanced cases with half zero and half one strings of output. The total number of output strings is $2(2^2) = 16$, but only the relevant ones are shown. The strings of outputs left out have a nonconstant or nonbalanced number of zeros and ones. Utilizing the formula given above with $n = 2$, for the number of balanced output strings, $(2^2)!/[2!\ 2!] = 4!/4 = 6$.

00	0	1	0	0	0	1	1	1
01	0	1	1	1	0	1	0	0
10	0	1	1	0	1	0	1	0
11	0	1	0	1	1	0	0	1
INPUT	CONSTANT		^	BALANCED				^.#

17.5 Quantum solution for the Deutsch-Jozsa problem

The solution of the Deutsch-Jozsa problem using quantum circuits involves only one quarry. It provides the solution in one shot The circuit to perform this solution is given in Fig. 17.4. The solution is very similar to the standard Deutsh problem. As can be seen from this figure, the input consists of $|0>@|0>@\ldots@|0> = |00\ldots0>$, n-fold tensor of control qubits, and a single $|1>$ as the target input. After an application of Hadamard transforms, the qubit $|v1>$ is obtained. This result is given by $|v1> = 1/2^{(n+1)/2}|+ + \ldots + - >, = 1/2^{n/2}$ sum, $\sum_x |x>@|->$, for x in $\{0,1\}^n$ $|x>@|->$. In the next step, before the oracle is utilized, notice that $|x->$ acts like an eigenvector for the oracle U; indeed, consider the qubit pair $|x->$. Then $U|x->\ = |x-(+)\ f(x)> = 1/2^{1/2}\ [|x\ 0\ (+)\ f(x)> - |x\ 1\ (+)\ f(x)>] = 1/2^{1/2}\ (-1)^{f(x)}\ |x|->$. The eigenvalue corresponding to this eigenvector is $1/2^{1/2}\ (-1)^{f(x)}$ (Jozsa, 1994).

At qubit $|v2>$ after the oracle is applied, it follows that $|v2>$ equals for x in $\{0,1\}^n$ of the sum, $\sum\ 1/2^{n/2}\ [(-1)^{f(x)}\ |x>|->]$. Before $|v3>$ is found, note that the single Hadamard operator acting on $x = |0>$ or $|1>$ can be written as $H|x> = 1/2^{1/2}$ sum, $\sum(-1)^{xz}|z>$, for z in $\{0,1\}$. When x is a tensor product of basis states, then a similar

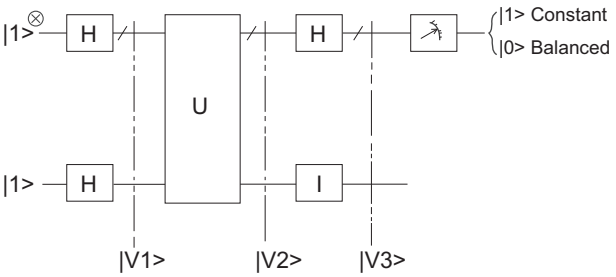


FIGURE 17.4 Deutsch-Jozsa algorithm.

formula holds H@n|x> =$1/2^{n/2}$ sum, $\sum$ $(-1)^{<x,y>}$ |z>, for z in {0,1}^{n}. Here, <x, z> is bitwise inner product mod 2.

At qubit |v3>, this is after the n-fold Hadamard gate is applied to |x> and the identity to |−>. In this case, |v3> equals for x in {0,1}^{n} $1/2^n$ sum, $\sum$ [for z in {0,1}^{n}, sum, $\sum((-1)^{f(x)}$ $(-1)^{<x,y>}$ |z>@|−>)]. These are the final output qubits. Lastly a measurement is performed on the standard basis on the control qubits |z>. The probability of finding that |z> = |0> involves the amplitude absolute squared. This probability is |for x in {0,1}^{n}, $1/2^n$ sum $\sum(-1)^{f(x)}$ |^2. For f constant, the probability summed yields one. For f balanced since the partial sums alternate, zero is obtained. In any case, one execution cycle is employed in yielding this result.

17.6 Grover search problem

The Grover problem is an unstructured search problem involving a positive integer number N = 2^n of items. The underlying Hilbert space for this application is C^N. A typical structure for this searching problem is, for instance, f: {0, 1}^{n} → {0, 1}, and say that only one instance of the output yields a one; all the rest are zero. The objective is to find the item, x*, that is the string of length n, such that f (x*) = 1.

Example 17.2:

The Grover problem, in this case, is to find number six where N = 2^3 and f: {0, 1}^{3} → {0, 1}. In binary representation if x* = 110, then f (x*) = 1, and this is the sought-after answer. Otherwise, f (x) = 0. It may be considered that for each binary representation of integers 0 through 7, a flag is attached with a zero for all strings except for 110; it has a flag with a one. Below, these strings are provided in order; in general, this is not the case; the order might be random. Moreover, in general, the length of each string is positive integer n.

0	0	0	0	0	0	1	0
000	001	010	011	100	101	110	111
0	1	2	3	4	5	6	7

It is convenient sometimes to write the tensor |110> as |6>, using binary number system.#

Employing conventional computers, N queries are needed to solve this problem. In all cases, this number is a guarantee of finding the string labeled by the numeral one. That is, a search through every single item is needed to find a solution. This is similar to knowing a person's telephone number and trying to find the owner of that number going through a telephone book. When a random searching procedure is employed in solving this problem, about N/2 queries are only required. However, using quantum algorithms (Grover, 1996), only $N^{1/2}$ queries need to be made to determine the string labeled one. So, if N is about a million, instead of using brute force, by employing a quantum algorithm only a thousand applications are needed to solve this search problem. As mentioned previously, the best quantum computing can do is of order $N^{1/2}$ complexity when order N complexity is needed in conventional computing. The Grover algorithm shows this result is sharp.

In general, quantum computing is referred to as bounded error quantum probability (BQP) computing. This class of computing includes conventional P, that is, solving a problem in polynomial time. It also includes a portion of nondeterministic polynomial (NP) computing, such as factoring. The NP complexity class is where verification of solutions in polynomial time can occur, but the solution itself cannot be found in polynomial time. BQP does not include NP-hard or NP-complete problems. NP-complete includes NP and other problems in NP with polynomial overhead that can be translated to it (Encyclopedia Britannica).

17.7 Solution to the Grover search problem

The Grover algorithm begins with a uniform superposition of inputs. There are N inputs each with state $|0>$, by applying a Hadamard transform yielding $|w> = 1/N^{1/2} \sum_{x=0}^{N-1} |x>$, where again x is a string of length n of zeros or ones. In binary, this string represents the integer number k lying in [0, N −1]. So the probability amplitude of each of these tensors is $1/N^{1/2}$. If a measurement was performed, one of these tensors would be found with probability 1/N. This is true for every x including x*. The next step in Grover algorithm is the oracle; it will negate the sign of x*.

The solution to the Grover searching problem involves an oracle, U_f, which is a unitary gate. This gate has the property of flipping the sign, that is, negating an input vector $|x>$ whenever the Boolean function f (x) = 1. Thus, $U_f|x> = (-1)^{f(x)}|x> = -|x>$ when f (x) = 1 and equals $|x>$ whenever f (x) = 0. The objective is to find the unique x* such that f (x*) = 1, by using the oracle. In matrix representation, an N by N diagonal matrix would represent the oracle. On the main diagonal, there exist all ones for this matrix, except for the location corresponding to x*, where a value minus one would appear. An application of this matrix reflects the entry x* relative to all other entries x. The next step is to use amplitude amplification, explained in Section 16.10 and illustrated next. It increases the probability amplitude for x* and decreases the probability amplitude of all other elements x by a small amount. This follows because N is large and the total probability of all entries must equal one.

The amplitude amplification algorithm involves U_f as well as the reflection about $|w>$ operation, $R_W = (2|w><w| - I)$. Here I is the identity operation along with pairs of reflection, operators that is R_W preceded by U_f. These two operations performed in sequence will yield a rotation in two dimensions. In the Grover search algorithm, the two vectors $|x^*>$ and $|w>$ span a two-dimensional subspace in C^N. These two vectors are almost perpendicular since the inner product $<x^*, w> = 1/N^{1/2}$, and N is large. A vector $|s>$ can be found by subtracting $|x^*>$ from $|w>$, and rescaling. This results with $|s>$ orthogonal to $|x^*>$. Refer to Fig. 17.5A. In this diagram, there exist three vectors: along the vertical axis is $|x^*>$, perpendicular to $|x^*>$ is $|s>$, and at an angle θ above $|s>$ is the original uniform superposition state $|w>$. Here $\cos(\theta) = <s, w>$. Note that $|w>$ is comprised of a sum of $N = 2^n$ states, each with probability amplitude $1/N^{1/2}$. Also, $|s>$ is comprised of $2^n - 1$ states, each with probability amplitude $1/(N-1)^{1/2}$. Thus, the inner product $<s, w>$ is equal to $1/[N^{1/2}(N-1)^{1/2}][2^n - 1]$; this implies that $\cos(\theta) = [(2^n - 1)/2^n]^{1/2}$;

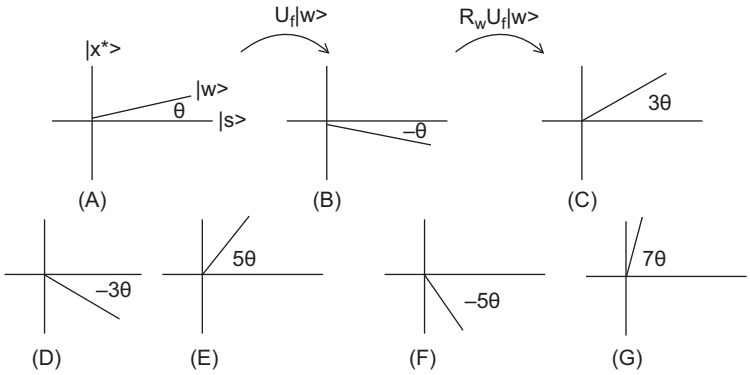


FIGURE 17.5 Grover reflection operations. (A) Starting position, (B) Application of Oracle, (C) Reflection applied, (D) Application of Oracle again, (E) Reflection applied again, (F) Oracle applied, (G) Reflection applied.

consequently, $\sin(\theta) = [1/2^n]^{1/2}$. For n large, θ is about equal to $1/2^{n/2}$, since the sine of a small angle is approximately equal to that angle.

By first applying the oracle U_f to $|w>$, a new vector is found by the reflection about the $|s>$ axis. The resulting vector, $U_f\ |w>$, is illustrated in Fig. 17.5B, Finally, utilizing the actual reflection operator R_W on $U_f\ |w>$ yields a new vector, $R_W\ U_f\ |w>$, which is located at an angle 3 θ above the $|s>$ axis. The angle follows since $R_W\ U_f\ |w>$ is at an angle 2 θ above $U_f\ |w>$. See Fig. 17.5C. Next, the Fig. 17.5D illustrates $U_f\ R_W\ U_f\ |w>$, where the oracle is applied again. Then if the oracle is followed by the reflection operator R_W, a new vector $R_W\ U_f\ R_W\ U_f\ |w> = [R_W\ U_f]^2\ |w>$ is found at an angle of 5 θ from the $|s>$ axis. This can be seen in Fig. 17.5E. The next diagram, Fig. 17.5F, shows the reflection along the x-axis, which is followed by another application of R_W illustrated in Fig. 17.5G. These vectors have angles that seem to be monotonically increasing in a counterclockwise manner to ninety degrees, which is the location of x*. However, stopping conditions are needed. This makes sure that enough pairs of operators $R_W\ U_f$ are applied to get close enough to x*, but not to wildly overshoot it.

In the Grover search algorithm, the two vectors $|x*>$ and $|s>$ are ON and span the plane. From geometry or using the inner product, $<s, w> = \cos(\theta)$ and $<x*, w> = \sin(\theta)$. As mentioned earlier, pairs of reflections $R_W\ U_f$ applied to the initial superposition vector $|w>$ enable the resulting vector to approach the solution vector, $|x*>$. Let $\Psi^r = (R_W U_f)^r |w>$ be the resulting vector by employing r applications of the reflection pair. So, for $r = 0$, ψ_0 makes an angle of θ from $|s>$. For $r = 1$, ψ_1 makes an angle of 3 θ from $|s>$. For $r = 2$, ψ_2 makes an angle of 5θ from $|s>$. The object now is to find r where ψ_r makes an angle of $(2\ r+1)\theta$ from $|s>$. Again from the geometry, or using the inner product, $<s, \psi_r> = \cos((2\ r\ +1)\ \theta)$ and $<x*, \psi_r> = \sin((2\ r\ +1)\ \theta)$. The objective is to get $|\psi_r>$ as close to $|x*>$ as possible, and the probability of this happening is $\sin^2(2\ r\ +1)\ \theta)$. Since $\sin^2(2\ r\ +1)\ \theta)$ is maximum for $(2\ r\ +1)\ \theta = \pi/2$, using $\theta = 1/2^{n/2}$, as found earlier, implies that $r = [(\pi/2)\ 2^{n/2} - 1]\ /\ 2$. So about $(\pi/4)\ 2^{n/2}$ is the value for r. To find x, $N^{1/2}$ tries are all that are needed. If there are k items with value 1, then $(N/k)^{1/2}$ tries are all that are needed. Quantum circuit illustrations of the refection operations used in Grover's algorithm can be found in Nielsen and Chiangmai (2000).

Example 17.3:

The solution to the Grover problem, in this case, is to find number 6 where $N = 2^3$, and f: $\{0, 1\}^3 \rightarrow \{0, 1\}$. In binary representation if $x^* = 110$, then $f(x^*) = 1$, and this is the sought-after answer. It is assumed that states $|0>$ are inputted and Hadamard transforms provide the uniform superposition state $|w> = 1/8^{1/2} \sum_{x=0}^{8-1} |x>$. As before, each x is a string of length three illustrated in Example 17.2. The next step is to find the state $|s>$, which is orthogonal to $|x^*> = |110>$. As mentioned earlier, this vector is found by subtracting $|w>$ by $|x^*>$ and rescaling to make the resulting vector have length one. So, $|s> = 1/7^{1/2} \sum_{x=0,not6}^{8-1} |x>$. The number r, of reflection pairs $(R_W U_f)^r$, to apply is given by $r = \pi/42^{3/2}$, which is about 2. Also $\theta = 1/2^{n/2}$, which in this case is about 0.35. Accordingly, the first application of the pair $(R_W\ U_f)$ on $|w>$ increases the angle from $|s>$ to $R_W\ U_f|w>$ to about 0.95. This is illustrated in Fig. 17.5. The next application of this pair of reflections provides an angle between $|s>$ and $(R_W U_f)^2|w>$ of about 1.75, which is close to $\pi/2$.#

17.8 The Shor's cryptography problem from an algebraic view

The Shor's problem has the objective of breaking the code in conventional public key cryptography. Just about all cryptography methods utilize a very large integer N, which is generated as a product of two large prime numbers. Although N might be known, the prime factors p and q are extremely difficult to find. The objective of Shor's problem is to find these primes p and q, such that $p \cdot q. = N$. Determining these factors is the critical point in all conventional cryptography methods. It is employed in RSA and in all current methods such as Diffie–Hellman public and private key exchange, as well as more advanced methods, including elliptic curve (EC) cryptography. In all these cases, a large amount of algebra is utilized along with number theory for developing and describing these cryptography methods. However, the crucial step is always to find the two primes whose product is N. Factoring N by conventional methods involves actual multiplication using prime numbers one at a time, p from 2 up to about $N^{1/2}$ and would take about $N^{1/2}$ steps. When N is represented by d decimal digits, the order of difficulty becomes exponential in d. The best-known number theoretic methods involving number field sieves provide a complexity of an exponential raised to $d^{1/3}$. Shor's algorithm has a runtime complexity, which is a polynomial in d.

Shor's contribution (Shor, 1997) is mainly that of converting this factoring problem into a problem of finding the period of a generating element in a cyclic group. The value of the period is also called the order of the group. In a previous section, the cyclic group Z_n was introduced in the context of wraparound signals. Now, it will be described more formally in the direction of presenting the Shor periodicity problem. In the following, there is a deliberate avoidance of using equivalence classes until a partition is described at the end of this section. The algebra and number theory background will be outlined now. Beginning with positive integer $N > 1$, the Euler's totient function, ϕ (N) (Landau, 1966), is the cardinality of the set of all integers between 1 and N inclusive, which are coprime or

relatively prime with N. The important case is when N is a product of two primes p and q, that is, $N = p \cdot q$. The Euler phi totient set will be described using the carrier set, $\Phi(N) = \{n$, such that n is an integer less than or equal to N, $n > 0$, $\gcd(n, N) = 1\}$. As usual, gcd is the greatest common divisor, and this set consists of integers n, which are relatively prime or coprime with N.

Importantly, the set of elements in $\Phi(N)$ form an abelian group-type algebraic structure under multiplication modular N. It is called Euler's totient group in the following presentation. Closure for the multiplication operation modulo N occurs since for x and y in $\Phi(N)$, $x \cdot y$ is also in $\Phi(N)$, since the gcd $(x \cdot y, N.) = 1$. The associative and commutative laws follow from similar properties of the integers. The number one is the identity in $\Phi(N)$, and the multiplicative inverse of x mod N is the integer y in $\Phi(N)$, such that $x \cdot y = 1$ mod N. The proof of the inverse property depends on Bezout's lemma (Ore, 1948). An important result derived from $\phi(N)$ is that the cardinality of $\Phi(N)$, that is, the cardinality of the Euler's totient group, is the product $(p-1)(q-1)$ where $p \cdot q = N$, and p and q are primes.

Example 17.4:

If $N = 14$, then the career set is given by $\Phi(14) = \{1, 3, 5, 9, 11, 13\}$. Note that 9 is in $\Phi(14)$ because the largest integer dividing both 9 and 14 is 1, whereas 8 is not in this set because $\gcd(8, 14) = 2$. Moreover, illustrating the abelian multiplicative group structure, observe that $9 \cdot 3 \bmod 14 = 13$, since 27 divided by 14 has a remainder of 13. Also, 1/11 mod 14 is the same as solving the problem, $11 \cdot ? = 1$ mod 14, where "?" must be in the carrier set $\Phi(14)$. The solution is $1/11 = ? = 9$, because $9 \cdot 11 = 99$, and when divided by 14, a remainder of 1 is left. The identity function for this abelian group is 1. Finally, the number of elements in $\Phi(14)$ is $(7-1)(2-1) =$ six, this is the value of the totient function.#

The cyclic group needed and mentioned earlier is a subgroup G_a of $\Phi(N)$. A cyclic group is generated by using an element a from the Euler totient group. The cyclic group is multiplicative mod N and consists of $\{1, a^1, a^2, \ldots, a^r\}$. The smallest positive integer r such that $a^r = 1$ is most important because r is called the period of a, or the order of the cyclic subgroup. The value a is called a generator for the cyclic group G_a. The most boring case is if $a = 1$, then by default the period is $r = 0$.

Example 17.5:

For $N = 15$, the Euler's totient group $\Phi(15) = \{1, 2, 4, 7, 8, 11, 13, 14\}$. This is a very important example because it was used for the first time to experimentally solve Shor's problem using quantum technology (Vandersypen et al., 2001). To illustrate the periodicity, let $a = 4$, then $G_4 = \{1, 4\}$, and the period of a is two, because $4^2 = 16$, but this value mod 15 is 1 since the remainder of 16 when divided by 15 is 1. Consider $G_7 = \{1, 7, 4, 13\}$. In this case, the period is four. It is the number of elements in this subgroup of $\Phi(15)$.

To illustrate the periodicity, like in a graph of a cosine wave or a sine wave, consider the function $f: Z_+ \rightarrow G_a$, for some nonidentity element a in $\Phi(N)$. Here, Z_+ are all the nonnegative integers, and $f(n) = a^n$. It would be useful to graph n versus f(n).

Example 17.6:

Again refer to Example 17.5. This time, use $f(n) = 7^n$. Then f: $Z_+ \rightarrow G_7$, in $\Phi(15)$, is provided below with the nonnegative integers, n given above and the corresponding values f (n) given below:

0	1	2	3	4	5	6	7	8	9	10...
1	7	4	13	1	7	4	13	1	7	4....

In the above diagram, it becomes apparent that the period is four; that is, the pattern repeats every fourth time and forms a partition of Z_+, resulting in the creation of four equivalence classes. #

17.9 Solution to the Shor's problem

The solution to the cryptography problem involving integer factorization is solved by Shor with quantum computations in polynomial time. As mentioned earlier, this is exponentially faster than conventional solutions. The procedure is roughly performed in two stages: First, similar to the Deutsh-Jozsa algorithm, quantum parallelism and constructive interference are utilized in determining the periodicity of a function, rather than seeing if it is balanced or not. Next, the period for a generator a of the cyclic group is found using the QFT. From an engineering perspective, QFT is used in determining the frequency = 1/ period. In any case, it is the latter procedure that produces the exponential speed up in using Shor's algorithm. Moreover, in actuality, quantum phase estimation is employed with unitary operations. Randomly choosing eigenvalues for these operators leads to determining the desired period r.

The actual steps, proposed by Shor that are needed in determining a factor p or q of N, are summarized below. They will be described in terms of the Euler's totient group, $\Phi(N)$; therefore, all computations are performed modulo N, for N a very large positive integer. Also, to set the stage for cryptography-type operations, assume that $N = p \cdot q$, where p and q are prime. The procedure is to do the steps in order:

1) Choose a random integer a in (1, N); a lucky guess would be if the value does not belong to the Euler totient group, $\Phi(N)$.
2) Compute k = gcd(a, N); Euclid's algorithm is a standard procedure.
3) If k differs from one, it must be p or q, and the search ends otherwise.
4) Find the period r for the chosen value a, i.e., find the order of the cyclic group G_a.
5) If r is odd, choose a different element a, in the interval (1, N), and repeat steps starting with the computation in step 2) otherwise.
6) The product $(a^{r/2}+1) \cdot (a^{r/2}-1) = 0$, and therefore, the factors of N are gcd $(a^{r/2}+1, N)$ and gcd $(a^{r/2}-1, N)$. These two numbers may be p and q, but with low probability, they could be 1 and N. These are factors. However, if the latter occurs, begin with step 1) and repeat the procedure.

Example 17.7:

Again let N = 15. Note that in Example 17.5 the Euler's totient group is Φ(15) = {1, 2, 4, 7, 8, 11, 13, 14}. As mentioned earlier, a lucky guess would be a number, a not in Φ(15). Shor's steps to find the prime divisors of N are as follows:

1) Say that a = 6, a number not in Φ(15) is chosen.
2) Compute gcd (6, 15), Euclids algorithm: $15^* = 6^* \times 2 + 3^*$
$6^* = 3^* \times 2 + 0$, yields $\gcd(6, 15) = 3$
3) Since k = 3, the procedure ends with p = 3, and an easy computation will always provide the other factor; in this case, it is q = 5.#

Example 17.8:

Again let N = 15; the Euler's totient group is Φ(15) = {1, 2, 4, 7, 8, 11, 13, 14}.

1) Say that a = 4 is chosen.
2) Compute gcd (4, 15), Euclid's algorithm: $15^* = 4^* \times 3 + 3^*$
$4^* = 3^* \times 1 + 1$, yields $\gcd(4, 15) = 1$
3) Since k = 1, it must continue in order.
4) To find the period of a, or the order of the cyclic group $G_4 = \{1, 4\}$, it is r = 2.
5) The value r is not odd.
6) The product (4+1) • (4−1) = 0, because 15 divided by 15 has remainder zero. In this case, gcd(4−1, 15) = 3 and gcd (4+1, 15) = 5. So these are the prime factors of 15.#

Actual quantum circuits implementing algorithms similar to those specified earlier can be found in Nielsen and Chiangmai (2000) and Beauregard (2002).

17.10 Elliptic curve cryptography

Elliptic curve cryptography (ECC) employs Galois fields, F_q, otherwise known as finite fields. These fields are described below. ECC is much more efficient and secure than other present cryptography systems. For instance, in conventional information transmission of keys with 3072 bits, as employed in RSA, then using ECC only 256 bits are needed to yield the same level of security. In top secret information transfer, ECC keys are 384 bits long, while RSA would require 7680 bits to provide the same level of security. The result of using ECC is an increase in speed and security.

ECC is utilized in numerous cryptocurrency applications involving blockchain technology. These include Bitcoin and Ethereum; they use secp256k1 (Cook, 2018). In blockchain applications, and in general, a known special point on the EC is found. It is called the base point or primitive element. The actual point is (x,y) and is located on the EC. Therefore, this is an element of F_q x F_q, the Cartesian product of F_q with itself. This tuple pair is shared by all users. It is called the shared key or public key. Specifically, for crypto coin applications, it is represented by 130 hexadecimal digits. In all crypto-type applications, point addition and the doubling operation are utilized and described below. For Bitcoin

and Ethereum, the ECC is performed using the specific EC, $y^2 = x^3+7 \bmod p$. For this application, p is the very large prime number, $p = 2^{256}-232-977$. The private key is referred to as the order of the base point; it is an integer n. In these applications, the private key is represented with 60 hexadecimal digits. The private key n is used with the primitive element by adding (x,y), over and over again, n times. The operation is based on the addition rule for this commutative group described below. Doing this results in $n-1$ other point pairs on the EC. The value n is a hidden number specific to a single user. It is the private key. So in summary, the base point (x, y) is shared, along with the product of n multiplied by the base point (x, y), but not n itself. The value n is known as the secret key; it is never shared. As mentioned before, ECC executes operations using an underlying Galois field,. This field is summarized and illustrated below.

Galois fields exist only for $q = p^n$, where p is a prime number and n is a positive integer. These fields are denoted by GF(q) or by F_q. The order of the field or its cardinality is given by p^n. The characteristic of the field is p; it has the property that when adding any element in F_q, p times the result is zero. The nonzero elements of F_q form a multiplicative cyclic group G_a. As in any cyclic group, there is at least one generating element a, for which all powers of a produce all elements of G_a. In this case, a is also called a primitive element of F_q. In cryptography, the most important field is the prime field $F_p = Z/pZ$; however, in these applications the prime number p is very large. F_p consists of integers modulo prime p, with the usual addition and multiplication involving mod p arithmetic. As previously stated, the elements of F_p will be represented as integers in the carrier set {0, 1, 2, . . ., $p-1$}, but in actuality there are equivalence classes created. The elements in the carrier set act as leaders or representers for each equivalence class. As before, the equivalence classes form a partition of Z. Also see Example 17.6.

Example 17.9:

Consider the Galois field F_5. From the aforementioned description, the carrier set is {0, 1, 2, 3, 4}, and the order of the field is 5. The characteristic of the field is $p = 5$; so, for instance, adding the number 2 five times, by using 4 plus signs, gives 10 mod 5, but when dividing 10 by 5, a remainder of 0 is found. If the number $a = 3$ is used as a generating element for the cyclic group G_a, then $a^0 = 1$, $a^1 = 3$, $a^2 = 4$, and $a^3 = 2$. So G_3 has a carrier set {1, 2, 3, 4}, and $a = 3$ is also called a primitive element. For the Galois field F_5, utilizing G_3 causes the partition illustrated below for all the integers. In this diagram, the top row depicts all the integers n, and the bottom row depicts powers 3^n:

. . . −2	−1	0	1	2	3	4	5	6	7. . .
. . .4	2	1	3	4	2	1	3	4	2. . .

For instance, referring to the aforementioned diagram, $3^{-1} \bmod 5$ means that $3 \cdot (?) = 1 \bmod 5$, and $? = 2$ is the correct answer. Accordingly, -1 is in the same equivalence class as 3, 7, and so on with period $p-1 = 4$.#

Although, at present cryptography is usually performed in F_p, for p prime, for completeness sake, it is interesting to provide an example of a Galois field F_q, where $q = p^m$, with $m > 1$. Such Galois fields might be employed in the future.

Example 17.10:

In this example, the carrier set for the Galois field consists of four elements, $F_4 = \{0, 1, a, a + 1\}$. The six operations needed for a field structure are provided in an earlier section; these operators are named in Figure 1.1. To begin, the zero-ary operators are first the ZERO element, which is 0, and the ONE is 1. For the binary operations, the actual ADD operation is symbolized as $+$. The actual MULT operation is denoted by $\cdot$, or when applied to elements within the carrier set or anywhere else, these elements may be placed side by side. Actual operations are best explained by referring to the tables provided below. These tables are like matrices; however, the elements within the carrier set are located in the top row and the leftmost column for each table. Since the field structure is abelian, both of these tables are symmetric about the main diagonal.

+	0	1	a	$1 + a$
0	0	1	a	$1 + a$
1	1	0	$1 + a$	a
a	a	$1 + a$	0	1
$1 + a$	$1 + a$	a	1	0

$\cdot$	0	1	a	$1 + a$
0	0	0	0	0
1	0	1	a	$1 + a$
a	0	a	$1 + a$	1
$1 + a$	0	$1 + a$	1	a.

For instance, in the addition table using the bottom entry on the left column, and the rightmost entry on the top, shows that $(1+a)+(1+a) = 0$. Using the same entries in the multiplication table gives $(1+a) \cdot (1+a) = a$. On the other hand, $(1 + a) \cdot (1 + a) = 1+2a+a^2$, but from the same table $a^2 = a+1$, so $(1+a) \cdot (1+a) = 2+3a$; however, $2 = 0$, from the addition table, $1+1 = 0$, and $2a = 0$, from that same table; consequently, $(1+a) \cdot (1+a) = a$, as previously mentioned. Before unary operations are described, it should be mentioned that the carrier set for $F_4 = \{0, 1, a, a + 1\}$. Any other number such as 2 or a^2 can be considered to be in an equivalence class where the coset leaders are in the carrier set. So without using the brackets, [] for cosets, we just let $2 = 0$ and $a^2 = a+1$.

The unary operations follow again from these tables. For MINUS, the symbol $-$ is utilized. From the addition table, to find $-x$, find x on the left column. Then using the row containing the value x, find in that row the value 0. Now, the column containing that zero also contains the value of $-x$ as the topmost entry. For instance, $-a = a$. In other words, $-a$ is whatever is added to a to yield zero. So going to the addition table and finding a in the left column, in the row containing a the number zero should be found. In the column for which the zero element was found, using the top element shows that $-a = a$. The final unary operation is a partial operation; it is INV and denoted by /, or $(.)^{-1}$, etc. Again, use the multiplication table, but this time to find $1/x = x^{-1}$, where x is not zero. Use the left column to find x, and then in that row, find the value 1. The solution is found using the top value in the column containing the one. For instance, $1/a = 1 + a$. Do not forget that a^{-1} means that $a \cdot (?) = 1$; indeed, $a \cdot (1+a) = a+a^2 = a+1+a = 1$. All the equational identities hold. For a field structure, some identities were illustrated earlier, and since the field is finite, each identity can be easily checked. #

Example 17.11:

Referring to the previous example, using F_4, all the nonzero elements form a cyclic group isomorphic to Z_3. Here, $G_a = \{1, a, 1 + a\} = \{1, a, a^2\}$. #

17.11 MSA of elliptic curve over a finite field

ECs are presented in several sections in this text. The underlying field structure was the complex field, the real field, as well as the rational number field. Also, they were described in projective space. Moreover, it was shown that certain points on the curve can be used in generating an Abelian group under specified addition, minus, and zero operations. This fact illustrates the concept of an Abelian variety where the group is described algebraically even though the operations are motivated by geometric considerations. Moreover, the associative law was shown to hold for EC group structures in Appendix A.3. Additionally, the point at infinity, ∞, is the identity element within the group. Accordingly, $\infty+(x, y) = (x, y)+\infty = (x, y)$. Including the point at infinity makes the algebraic structure become a pointed set. The carrier set for the EC is a commutative group structure including ∞, and the carrier set for the underlying field is a finite field. Now, the ECs are be illustrated within a Galois field structure, beginning with an example.

Example 17.12:

Consider the EC: $y^2 = x^3+2x+12$, over the Galois field F_{17}. The objective in this example is only to illustrate how a point pair is found on this curve. The point that is found could be a base point and used as a shared key in cryptography applications. For instance, if $x = 16$, then substituting into the EC gives $y^2 = 4096+32+12 = 4140$. Since the underlying field is mod 17, the number 4140 must be mapped to a number in the carrier set for the field, {0, 1, 2, . . ., 16}. Elements in this set are also called coset leaders. That is, 4140 has to be divided by 17 and the smallest nonnegative remainder must be found. As usual, the calculation is performed by Euclid's algorithm: $4140^* = 17^* \cdot 243+9^*$. Thus, the remainder is the desired number 9, so $y^2 = 9$. Accordingly, there are two corresponding point pairs on this curve: (16, 3) and (16,−3).#

The short Weierstrass normal form is employed throughout this section. This implies that the most general form of the EC will always be $y^2 = x^3+ax+b$, with a and b in F_q. The reason for not having a more general form is that the characteristic of the fields employed in the following do have a characteristic greater than 3. The characteristic for F_q is the smallest number of times the number one must be added to itself such that the result is zero. If there does not exist such a number, then the characteristic is defined to be zero. Note that from Example 17.10, in F_4, the characteristic is 2. Moreover, the characteristic of the real field, R, is zero.

Additionally, EC must be such that the equation $4a^3+27b^2$ is not zero. This guarantees that the EC is not singular, that is, there are no cusps, and as such the curve is smooth. For the EC, the partial derivatives should not be equal to zero simultaneously. Also, the curve should be simple, that is, it does not intersect itself. Illustrations of different EC are given in Fig. 17.6. As illustrated, there exist essentially two types of EC; the common feature is that the curves are symmetric about the x-axis. Additionally, all curves intersect the x-axis in one or three locations. Illustrations, not good-curves have no line segments.

Although the Abelian group structure and group operations for an ECC are the most important, in ECC there are two operations that are dominant in applications. These are

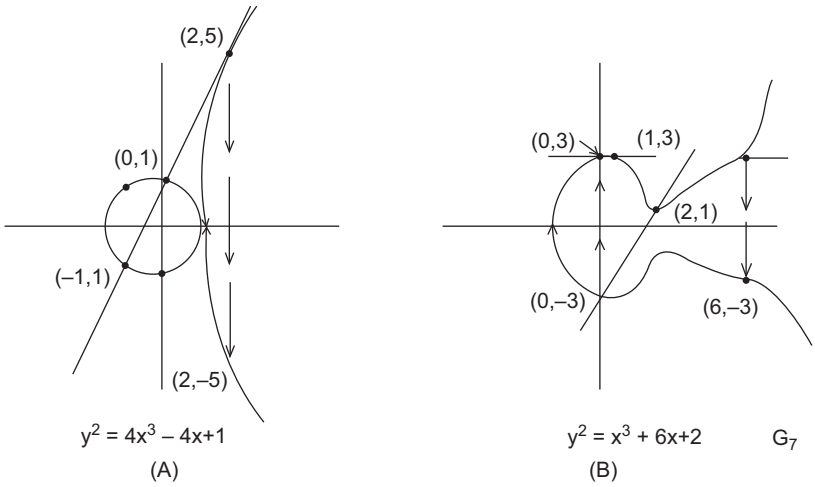


FIGURE 17.6 Types of elliptic curves.

the point or cord addition and the point doubling operation. Both methods arise from using basic calculus techniques to describe the geometric approach. For point addition, use two points (x1, y1) and (x2, y2), both on the EC and connected by a straight nonvertical line on the EC. Then, a third point (x3, − y3) lies on this same line, and on the EC as well. When this point is reflected about the x-axis, then the result (x3, y3) is called the point addition for the first two points. Next, the doubling operation on a point occurs by finding a nonvertical tangent line to a point (x1, y1) on the EC. This line will intersect the EC at a point whose reflection about the x-axis yields (x3, y3). The result is called the doubling of the original tangent point. The point addition and doubling operations are described graphically in detail, in Appendix A.3. Now these operations are found from basic calculus and are presented algebraically relative to the short Weierstrass normal form of EC: $y^2 = x^3 + ax + b$.

For the cord addition or the two-point addition:

The slope is $s = (y2 - y1)/(x2 - x1)$.
The x coordinate is $x3 = s^2 - x1 - x2$.
The y coordinate is $y3 = s\,(x1 - x3) - y1$.

For the doubling or the double-tangent method:

The slope is $s = (3\,x1^2 + a)/(2\,y1)$.
The x coordinate is $x3 = s^2 - x1 - x1$.
The y coordinate is $y3 = s\,(x1 - x3) - y1$.

An example will illustrate the use of these equations.

Example 17.13:

Consider the EC, $y^2 = x^3 + 6\,x + 2$, over F_7. Here, $a = 6$, and $b = 2$. First, notice that $4a^3 + 27b^2 = 864 + 108 = 972$. Now dividing the quantity by 7 and retaining the smallest

nonnegative remainder gives 6 in F_7, so it is not zero, and the EC is not singular. First, the two-point cord method will be employed. Use (0, 3) and (1, 3) as the two distinct point pairs to be added together. These points are on the EC. Using (0, 3) and substituting into the EC shows that $y^2 = 2$. This square has solutions $y = 3$, as well as $y = 4$, because squaring each of these results in 2, thus providing the unique number in the carrier set F_7. In any case, using these two points, that is, (0, 3) and (1, 3), yields a slope $s = 0$, $x3 = -1 = 6$, $y3 = -3$. As a result, point addition yields (6, −3), and this point is on the curve. Substituting into the EC gives $(-3)^2 = 2$, and $y^2 = 6^3+6 \cdot 6+2 = 216+36+2 = 254 = 36 \cdot 7+4 = 2$. Next, the tangent-doubling method will be illustrated using the point (2, 1) on the EC. A quick substitution into the EC shows that $1^2 = 2^3+6 \cdot 2+2 = 22 = 1$. The slope is $s = (3 \cdot 2^2+6)/(2) = 9 = 2$. Therefore, the x coordinate is $x3 = 2^2-2-2 = 0$. The y coordinate is $y3 = 2\ (2-0)-1 = 3$. Accordingly, the new point is (0, 3). As before, this point is on the EC. The EC in this example is illustrated in Fig. 17.6B. Here, the corresponding procedure for point addition and tangent point doubling is roughly illustrated graphically.#

In the general case, where the EC is $y^2 = x^3+ax+b$, a few algebraic computations not covered by the aforementioned equations will be described next. For point addition, using two points (x1, y1) and (x2, y2) connected by a straight line, which is a vertical line on the EC, the result is the neutral point at infinity, ∞. This can also be seen since in this case, (x2, y2) = − (x1, y1) = (x2, −y2), resulting in the neutral element, a third point (x3, y3) being ∞. The same is true for the tangent-doubling method: If the tangent line is vertical, the third point is ∞. Finally, it was mentioned before that when the point ∞ is added to any other point, z on the EC, including itself, the result is z.

Example 17.14:

Again consider the EC, $y^2 = x^3+6x+2$, over F_7. Refer to the diagram given in Fig. 17.6B. In this diagram, the two points (1, 3) and (1, −3) = − (1, 3) = (1, 4) are illustrated, and the vertical line that passes through these two points yields the point ∞.#

For an EC, E with Galois field F_p, the elements form an abelian group. The order of the group, that is, its cardinality E, must be known exactly, and this is the case in ECC. However, in general Hasse's theorem over mod p provides bounds for #E (Cassels, 1991). These are lower and upper bounds; specifically, $p+1-2p^{1/2}$ is less than or equal to E, which itself is less than or equal to $p+1+2p^{1/2}$. These bounds will be illustrated by way of an example.

Example 17.15:

Consider the EC, E: $y^2 = x^3 +1$, using the Galois field F_{11}. Since $p = 11$, Hasse's theorem shows that the cardinality of E, E, is an integer lying in the interval $[11+1-2 \cdot 11^{1/2}, 11+1+2 \cdot 11^{1/2}] = [5.4, 18.6]$. For this curve, the carrier set is E = {∞, (10, 0), (0,10), (0, 1), (9, 2), (9,9), (6, 3), (6, 8), (8,4), (8, 7), (3, 5), (3, 6)}. The actual cardinality is E = 12. To validate the point (0, 10), that is, to verify that this point is in E, note that $y^2 = 1$, when $x = 0$. If $y = 10$, $y^2 = 100$, and when this number is divided by 11, the least positive remainder is 1; note that $-10 = 1$. #

Getting back to ECC, the discrete log problem referred to as ECDLP utilizes an EC with a primitive element a, that is, a point pair (x, y) on the EC. This element generates all points on the EC using point-adding and tangent-doubling operations. In particular, the point $t = c+c+\ldots+c$ is n time hops on the EC. The starting point c is a public key c that is a point pair on the EC. The integer n is a secret or private key; it is an integer and is not known. However, the product of n with c, which is t, is known to all users. In actual applications like blockchain, the prime number utilized is very large, as seen previously. The cardinality for the carrier set of E, #E, is very large and so is n. Since n is only known to a private user, the value $n \cdot c$ is efficiently found using the algorithm specified below. However, knowing c and $n \cdot c$, it is almost impossible to find n. The use of quantum computers, as described in Section 17.9, makes breaking the code simpler.

The quick algorithm to find $n \cdot c$ requires a binary representation for the positive integer n. Here, the leftmost bit is the most significant and is 1. Starting with the next most significant bit and working to the least significant bit, if a one appears, then first a doubling tangent operation is performed on c; then this is followed by a two-point addition operation. If a zero appears in the binary representation, only a doubling tangent operation is employed. An example will illustrate the methodology.

Example 17.16:

Since n is known to the private user, $n \cdot c$ can be determined by the above algorithm. Here assume that $n = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 = 51$, and in binary, it is (1 1 0 0 1 1). Starting with the point pair c on the EC, doubling yields $2 \cdot c$, followed by point adding that gives $3 \cdot c$. The next binary digit is zero, so only a doubling is utilized, giving $6 \cdot c$. Again, the next digit is doubling, so $12 \cdot c$ is the result. The next digit is one, so both EC operations must be performed. First doubling gives $24 \cdot c$, and followed by point addition, this yields $25 \cdot c$. The final bit is one, accordingly doubling provides $50 \cdot c$, and point addition gives the final result, $51 \cdot c$.#

17.12 Diffie–Hellman EEC key exchange

The EC Diffie–Hellman key exchange is described next. It consists of two phases: the setup and the exchange. For the setup, there are two users, and needed are the parameters for ECC. These are a, b, p, and primitive point pair c = (x, y). In the second phase, the two parties 1 and 2 agree on EC and p. Party 1 chooses a random number that is a private key in {2, 3, ...#E −1}, call it d. Party 1 computes $D = d \cdot c = (x1,y1)$. Also party 2 randomly finds a number, f in {2, 3, ...#E −1} and computes $F = f \cdot c = (x2,y2)$; they exchange the publicly known values D and F. However, when received both multiply, using their secret number, and obtain $R = d \cdot F = f \cdot D$. The value R is private and enables correspondence using this address along with hashing techniques. Hashing techniques map using h, an arbitrary length signature-type string, into a large fixed-size binary string, length m. Here, $h: \{0, 1\}^* \rightarrow \{0, 1\}^m$. It is a one-way function, that is, knowing the original string, an algorithm is used to find the fixed-length binary string. However, using the binary string, it is almost impossible to regain the signature (Dorey et al., 2016).

References

Beauregard, S., 2002. Circuit for Shor's Algorithm Using 2n + 3 Qubits. Cornell U.
Cassels, J., 1991. Lectures on elliptic curves, London Mathematical Society Student Texts, 24. Cambridge University Press.
Cook, J., 2018. A tale of two elliptic curves. Elliptic curves secp256k1 and secp256r1, Internet.
Dorey, K., et al., 2016. Persistent Diffie-Hellman backdoors in TLS. Cryptology .
Grover, L., 1996. A fast quantum mechanical algorithm for database search. Proc 28th Annual ACM Meeting.
Jozsa, R., 1994. Fidelity for mixed quantum states. J. Mod. Opt. 41 (12).
Landau E., 1966. Foundations of Analysis, 4th ed. Chelsea.
Nielsen, M., Chiangmai, I., 2000. Quantum Computation and Quantum Information. Cambridge University Press.
Ore, O., 1948. Number Theory and its History. Dover books.
Shor, P., 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum. SIAM J. Comput. 26.
Vandersypen, et al., 2001. Experimental Realization of Shor's Quantum Factoring Algorithm Using Nuclear Magnetic Resonance. Nature.

Further reading

Bennett, C., Bernstein, E., Brassard, G., Vazirani, U., 1997. Strengths and weaknesses of quantum computing. SAIM J. Comput 26 (5), 1510–1523.
Landau, L., Lifshitz, E, 1996. Quantum Mechanics-Non-Relativistic. Pergamon Press.

This page intentionally left blank

CHAPTER

18

Machine learning and data mining

18.1 Quantum machine learning applications

Machine learning with adiabatic quantum computing (AQC) and quantum annealing (QA) employs both supervised and unsupervised methods to discover patterns and correlations directly from data sets. Global or subglobal optimization techniques accelerate training time in the learning process. A great book describing many applications of quantum in the data mining field is the text by Wittek (2014). The text was way before its time in illustrating quantum and data mining interfacing. Now many more applications in the machine learning environment employ quantum gate type computers as well as adiabatic computers. Several of these applications will be described below.

Facial patterns were obtained from a QA process (O'Malley et al., 2018). Additionally, and more recently, facial expression recognition using a quantum simulator was conducted (Mengoni, 2021). In this research, several machine learning techniques were employed. Quantum interference was exploited to perform classification similar to the k-nearest neighbor (KNN) classification algorithm. In this study, images were first identified using 68 point tuples (x, y) in R^2. This was followed by triangulation procedures and graph theoretic methods, leading to an adjacency matrix. The triangularization procedure is used in providing a noncomplete, meshed graph while not compromising information content. Subsequently, quantum states were defined by encoding tuples within adjacency vectors as amplitudes for quantum states. The conclusion from this work is that classical recognition methods provided better results when complete graphs were used. However, the quantum classifier yielded comparable results when the meshed graph was employed.

The Hyundai company collaborated with Ion Q quantum for autonomous vehicles (IonQ Staff, 2023). Also refer to the trapped ion technology, as developed by IonQ described in Section 6.7. Research in the past involved quantum computers in modeling battery performance to improve usage in self-driving cars. Currently, quantum computing is used to enhance safety and human engineering aspects in autonomous vehicles. Specific objectives involve blind spot warnings, emergency braking, lane control, as well as comfort within the vehicle to make traveling occur more leisurely.

IonQ quantum computing for Hyundai is primarily directed in two areas: image recognition of road signs and, second, object detection. In the former situation, forty-three distinct road signs were employed in a quantum machine learning simulator. Only 16 qubits

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00016-3

were utilized; however, this allowed estimates of how image recognition will improve utilizing more qubits. Fault-tolerant procedures were also investigated concerning object detection. In this case, the quantum computer had to discriminate objects based on these images. Boxes surrounding perceived objects within an image were used as an image segmentation procedure. The boxes were utilized in predicting the location of the object. However, several boxes could overlap and be of various sizes. The next task is to identify objects within each box.

AQC and QA methods are used for classification applications as well as training of Boltzmann machines. These machines are a type of NN (Liu and Rebentrost, 2018).

A decentralized feature extraction speech recognition system was developed using quantum convolutional neural nets (QCNNs) (Yang et al., 2021). A quantum circuit encoder as well as a quantum feature extractor were employed. Also a recurrent NN (RNN) was employed locally for determining the final recognition. The QCNN model showed greater than ninety-five percent accuracy in recognition. This article reported somewhat better results than utilizing previous centralized RNN systems. Spectrogram features are used as inputs to the quantum circuit layer. It is this quantum circuit that learns and encodes speech patterns. The front end of the QCNN performs encoding and organizing the input into two by two patches, thus allowing the quantum circuits to process the data. After the quantum circuits act on the data, a decoding process is entered. This procedure involves projecting qubits into a set of spanning quantum states.

The creation of a large data set for quantum machine learning applications was developed in Perrier et al. (2022). The QDataSet is made up of 52 publicly available data sets originating from simulations involving one or two qubits. Some data within the QDataSet contains noise, and some are without noise. Each data set is comprised of 10,000 samples useful in machine learning algorithm development. Specifically, this data set is intended to be utilized in machine learning applications such as quantum computation, quantum control, quantum spectroscopy, as well as in quantum tomography. These data sets are of particular importance in benchmarking algorithms in quantum control, as well as assisting in solutions to constrained optimization problems involving quantum data.

18.2 Learning types and data structures

Machine learning utilizes various data types. Vectors are one of the most basic data structures. Tuples within a vector constitute recorded values of attributes for designated entities, and possibly relationships. For instance, each patient in a doctor's office might have an associated vector. Here, for patient X, the vector might consist of tuple values for height, weight, blood pressure, mother's blood type, ..., smoker. Other data structures are matrices, images, videos, lists, sets, trees, graphs, strings, as well as macrostructures consisting of several of these data types. As an example, self-driving vehicles probably utilize all these structures with strings of natural language converted into navigation and guidance rules. Real-time image, sound, and video observations are sent to the deep learning machine. These are processed, and control signals are sent to appropriate control surfaces.

The conversion of actual observations into control signals often employs estimation and classification techniques. Some intuitive descriptions of the problems need solving by

deep learning machines, most often classification problems. In particular, binary classification entails a choice of determining for y, one of two random quantities. In this application, it is assumed that labels exist to make identifications. The random quantities might be locations, for instance, partitioning or separating a group of labeled objects into one of two segregated locations. In multiple classification, a choice of several random quantities needs to occur. Likewise, a partition of objects into several parts is a critical machine learning application. Labeling is an important problem in machine learning. Sometimes, labels may be clear and other times they may not be clear. In the latter case, objects possessing these labels are removed and often termed erasures. Additionally, sometimes the label contains more information, and it can be utilized in a beneficial manner. Regression is an extremely important machine learning tool. In this case, some a priori model is utilized in estimating or checking results of observations, and also for determining outliers. An intuitive description of different types of machine learning is provided as follows:

1) Online learning often consists of a pair of sequences (x_i, y_i), and the objective is to estimate y_i, in real time, when observing the x values. This is also called sequential filtering. When given such a sequence pair, and a past value of the y sequence, y_{i-k} is to be estimated; this is called sequential smoothing. Similarly, if a future value y_{i+k} is to be estimated, this is called sequential prediction. Estimation with missing x data is often called erasure estimation, interpolatory estimation, or estimation with missing variables.

2) Batch learning again uses a two-tuple pair (x_i, y_i), wherein the y values are to be estimated, but the x observation values are usually preassembled in a set. When these x values are chosen for model building, this is called active learning. If there are two x type observational sets somewhat correlated and in which a possible covariant shift correction was made, this results in cotraining.

18.3 Probably approximately correct learning and Vapnik-Chervonenkis dimension

Probably approximately correct (PAC) learning (Valiant, 1984) involves a function or a set of functions in some class. They are said to be learnable, which means that once trained on random samples under supervision, the function can be utilized in estimating test sets with a low error. Thus, the generalization error is small when the test data has the same distribution as the training data. In general, a good learner will learn close approximations to test data with high probability. More formally, let X be a set of all samples, also called the instance space. Associated with X is the set of all labels. The function that is to be learned is c: $X \rightarrow Y$ and is called a concept. Moreover, let H be a set of all concepts. Assume that a sample S of identical independent random variables of size m is to be used from a distribution D. The objective is to find an algorithm that produces a concept h in H, also known as the hypothesis. The basic idea is to determine or minimize the generalization error R, namely how far h is from c. This quantity will depend on the distribution D, as well as the sample S, that determines how h is chosen. Additionally, it will depend on what concept c is to be learned. Here, the generalized error R is defined as the probability that c and h differ, that is, R = P (c is not equal to h).

A concept class H is said to be PAC learnable whenever there exists an algorithm, for every distribution D on X, for every concept c in H, and for all small quantities, ϵ and δ in [0, 1], described next. Then the probability P (R is less than or equal to ϵ) is greater than or equal to $1 - \delta$. This must be true for all m greater than or equal to a polynomial ordered function in $1/\epsilon$, $1/\delta$. Additionally, taken into account are the computational costs of using x in X and representing target concept c. The major theorem due to Valiant involves a zero/one loss function in a finite number of models H, with empirical risk $R_n = 1/n \sum_{j=1}^{n} \text{sgn}(|f(x_j)$ not equal to $y_j|)$, where sgn (z) = 1 for z not zero and 0, otherwise. Also, $f_n = \min R_n(f)$, if min R(f) = 0, then for every n and $\epsilon > 0$, the probability $P(R(f_n) > \epsilon) <$card (H) $e^{-n\epsilon} = \delta$. The sgn function was used for a NN creating an approximation to continuous functions in Section 2.5.

An empirical method for estimating the capacity of a learning machine is described in Vapnik (2000). In this case, the training set consists of tuple pairs (x,w) where x is in X, a subset of R^n, and w is binary, in {0, 1}. The m point pairs are drawn from independent, identically distributed unknown distribution, P(x,w) = P(w | x) P(x), where P(x) describes regions of interest for the inputs. The input-output relationship is governed by P(x,w). Binary classification functions are described by the function f(x,a), where a is a parameter in parameter space A. The objective is to find a parameter a*, which minimizes the probability of error p(a) = E(w − f(x,a)). This quantity is actually an average value, where the expectation operation E is evaluated with respect to P(x,w).

Minimizing errors on the training set will be consistent iff the uniform convergence property holds (Vapnik and Chervonenkis, 1971), $\lim_{m->\infty} P[\sup_{'ainA}(p(a) - v(a)) > \epsilon] = 0$, where v(w) is calculated from the training set; it is the empirical mean absolute value, $v(w) = 1/m\{\sum_{i=1}^{m} |w_i - f(x, a)|\}$.

For a finite concept class, m needs to be only greater than or equal to a polynomial ordered function in $1/\epsilon$, $1/\delta$, as well as the log of the cardinality of H. For infinite concept classes, the Vapnik-Chervonenkis (VC) dimension can be used. The VC dimension involves the process of shattering. The vectors $x_1, x_2, \ldots, x_m$, in X are shattered by f(x,a), for a in A, if for any possible partition B_1 and B_0 of $x_1, x_2, \ldots, x_m$, there exists a function f(x,a*), such that f(x,a*) = 0, for x in B_0, and f(x,a*) = 1 for x in B_1. So to be shattered, all 2^m partitions must be employed. The VC dimension is the maximum number of vectors, $x_1, x_2, \ldots, x_k$, which can be shattered by f(x,a), a in A. In practice, a set of vectors is shattered when any arrangement of these vectors can be shattered. If no arrangement of all the objects can be shattered, then they are classified as nonshatterable.

Example 18.1:

The following example illustrates the shattering process. For brevity, the vectors $x_1, x_2, \ldots, x_m$ in X will be illustrated as points in R^2. It will be shown that for a single point m = 1, it can be shattered. Also two points, m = 2, can be shattered, and three points, m = 3, can be shattered. These are all illustrated in Fig. 18.1. The partition is created by f(x, a), an oriented affine line with parameters in A. The affine line is described by parameter weights w_x and w_y, along with a bias term b. Additionally, an indicator is employed for describing orientation. Here, an arrow on the affine manifold, pointing upward, indicates that to the right is B_1 and to the left is B_0. For (x,y) in R^2, the affine line is determined by

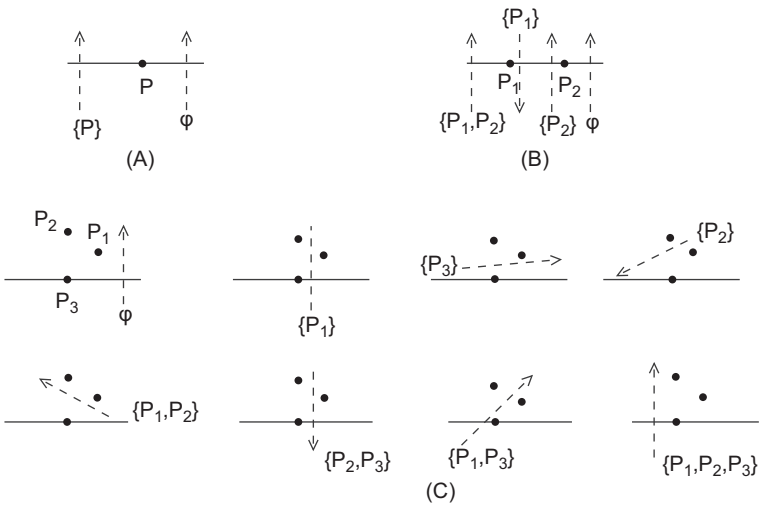


FIGURE 18.1 Shattering one, two, and three points by Affine manifold: (A) shattering a single point; (B) shattering two points; (C) shattering three points.

$w_x x + w_y y + b = 0$. Referring to Fig. 18.1A, a single point p is shattered by f since both subsets, the empty set ϕ and {p}, are created by an oriented affine line. If p = (0,0), then the affine line at x = 2, with an upward arrow, yields ϕ, because points within in the region B_0 are not counted; only points in B_1 are counted, that is, points to the right of the arrow. Again in the figure to the right is say x = −2, with an upward arrow, and so {p} is determined.

Similarly in Fig. 18.1B, two points are shattered by f. This is done in four steps; the most interesting step is where the left point has to be shattered by itself. Here the oriented affine line has an arrow going down. Thus all subsets consisting of two distinct points are created by f.

Finally Fig. 18.1C illustrates how three points are shattered by an oriented affine manifold. All eight subsets consisting of three points are created. For instance, referring to the figure where the set {p1, p3} is shattered, the affine line might be y = x+1/2, with an arrow upward.#

Analogous to the aforementioned example, it is shown (Vapnik, 2000) that in R^n exactly n+1 points can be shattered employing the partition B_1 and B_0 created by functions like f.

Example 18.2:

In R^3, four points can be arranged as vertices of a tetrahedron. An oriented affine plane can be employed to determine the $2^4 = 16$ desired subsets.#

Various other functions f, besides oriented affine manifolds, can be utilized in determining how large a set of points can be such that they are shattered.

Example 18.3:

Consider a set of four points in R^2. The function f will shatter these four points by determining all 16 subsets. Here the parameter set A will insure that f creates nonempty rectangles in the x, y plane such that the left edges of the rectangle are parallel to the y-axis. Accordingly, the top and bottom line segments of the rectangle are parallel to the x-axis.

Points within a rectangle correspond to B_1, and points outside of the rectangle correspond to B_0. Fig. 18.2 illustrates the arrangement of these four points to ensure that shattering exists. Four points taken one at a time yield four subsets. Four points taken two at a time yield six subsets. Four points taken three at a time yield four subsets. Finally, zero at a time and four at a time each yield one subset. In the referenced diagram, only two instances of shattering appear.#

Example 18.4:

Following Wittek (2014), the VC dimension could be infinite. To see this, let the set of points be all the odd integers on the x-axis in R^2. Any point, for instance, the point (1, 0) in the x-y plane, will be shattered by $\sin(\pi/2\ x)$ for x in [0, 2). Here B_1 is indicative of points below the sinusoid, and points above the sinusoidal are represented by B_0. To obtain all single-point sets among the odd integers, use parameters in A. Here use $\sum_{n=-\infty}^{\infty} |\sin(\pi/2\ x)|\ \chi_{[2n,2n+2)}$. To obtain all two adjacent odd point sets, use $\sum_{n=-\infty}^{\infty} |\sin(\pi/2\ x)|\ \chi_{[2n,2n+4)}$. For obtaining every two-point set containing every other odd point, use $\sum_{n=-\infty}^{\infty} \left[|\sin(\pi/2\ x)|\chi_{[2n,2n+2)} + |\sin(\pi/2\ x)|\chi_{[2n+4,2n+6)} + \right]$. And so on and so forth. This shows that all the odd integers are shattered by f, using the parameters in A.#

In the referenced papers by Vapnik, the VC dimension is used in predicting upper bounds for test errors in a classification model. These bounds involve independent identically distributed random variables, with the training and test data having the same underlying probability distribution functions.

18.4 Regression

Ridge regression is a least square technique mainly used when variables are thought to be highly correlated. In this case, the variance or least squared error is sometimes smaller when Ridge regression is employed rather than usual least squares. In the usual linear least squares regression for an overdetermined system, the pseudoinverse provides the solution. See Section 14.3, 14.4, where partial isometries are described. In this method, an observed $k \cdot 1$ vector y is projected onto the $k \cdot n$ design matrix H, with $k > n$ or $k = n$, and of full rank. An estimator of the form $x = B\ y$ is desired, where the model equation is $y = H\ x + u$, and u is a vector consisting of noise or slack variables. Moreover, it is assumed that nothing is known about the vector u, only that it is used to make the equations consistent. By projection methods, or by differentiating the least squares loss, the estimator is the pseudoinverse: $x = (H'H)^{-1}H'y$. The prime indicates the transpose operation. If more information is known about u, a better estimator could be employed.

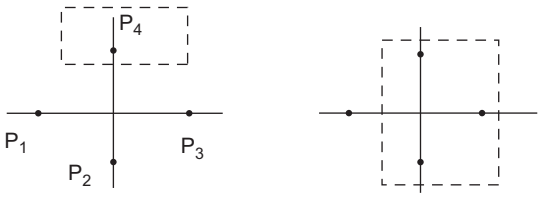


FIGURE 18.2 Shattering four points by a rectangle.

When the columns of H are correlated, the inverse may be inaccurate due to multicollinearity, which affects the condition number for the matrix H′H. In this case, a Ridge regression is often used. It too is a linear estimator; here: $x = (H'H + \lambda I)^{-1}H'y$.

The value $\lambda > 0$ is usually small; it is heuristically determined. Also, the identity matrix I is n • n. A simple example is used to illustrate the technique.

Example 18.5:

For $y = Hx + u$, use $y = (1.1\ 1\ 0.8)'$, $u = (u_1\ u_2\ u_3)$, and H =

$$\begin{vmatrix} 1 & 1 \\ 1 & .9 \\ 1 & 1 \end{vmatrix}$$

Then H′H, $(H'H)^{-1}$, and $(H'H)^{-1}H'$ are, respectively, given as follows:

$$\begin{vmatrix} 3 & 2.9 \\ 2.9 & 2.81 \end{vmatrix} \quad \begin{vmatrix} 140.5 & -145 \\ -145 & 150 \end{vmatrix} \quad \begin{vmatrix} -4.5 & 10 & -4.5 \\ 5 & -10 & 5 \end{vmatrix}$$

The linear least squares estimate is $x = (1.45\ -0.5)$. The residual, slack, or noise vector is $u = y - Hx = (1.1\ 1\ 0.8)' - (.95\ 1\ 0.95)' = (.6\ 0\ -0.15)'$.

If Ridge regression were used with $\lambda = 0.1$, just to illustrate the methodology, then $H'H + 0.1I$, $(H'H + 0.1I)^{-1}$, and $(H'H + 0.1\ I)^{-1}H'$ are, respectively, given as follows:

$$\begin{vmatrix} 3.1 & 2.9 \\ 2.9 & 2.91 \end{vmatrix} \quad \begin{vmatrix} 4.76 & -4.75 \\ -4.75 & 5.07 \end{vmatrix} \quad \begin{vmatrix} .01 & .5 & .01 \\ .32 & -.2 & .32 \end{vmatrix}.$$

The Ridge estimate $x = (.52\ -0.18)$. The residual, slack, or noise vector is $u = y - Hx = (1.1\ 1\ 0.8)' - (.34\ 0.35\ 0.34)' = (.75\ 0.65\ 0.46)'$.#

In the last example, ridge regression did not do well compared to regular least squares. However, as in many machine learning disciplines, heuristic parameters must be set. In this case, the value of λ could be changed, over and over again.

An interesting application is illustrated next when the feature space is an interval $[-1, 1]$ and the regression involves polynomials within this interval. Using a least squares estimator in the form of pseudoinverse, with observed data, $y = (x_1, x_2, \ldots, x_n)'$. In this case, the Vandermonde matrix V is utilized in the pseudoinverse construction, that is, H = V, and $x = (V'\ V)^{-1}V'\ y$, where V =

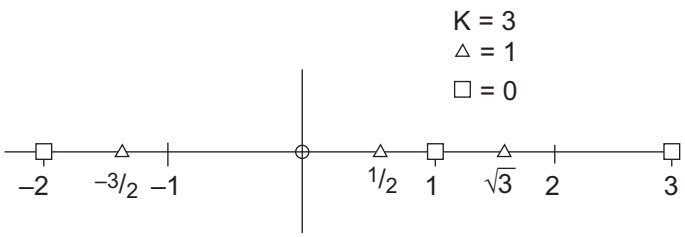
$$\begin{vmatrix} 1 & x_1 \ldots & x_1^m \\ 1 & x_2 \ldots & x_2^m \\ \bullet & \bullet & \bullet \\ 1 & x_n \ldots & x_n^m \end{vmatrix}$$

The Vandermonde matrix also appears in Example 8.21 in the context of cyclic vectors. However, in this regression application, the Vandermonde matrix often is unstable. A popular solution to increase stability is to utilize the Chebyshev orthogonal polynomials within the Vandermonde matrix. In Appendix A.7, the Chebyshev polynomials are shown to be an orthogonal set of polynomials since they are a solution to the Sturm-Liouville differential equation.

18.5 K-nearest neighbor classification

A most elementary example of machine learning is the KNN, similar to the K-means methodology described in Section 2.1. KNN is also a technique for classification and clustering. It is often used as an example of batch learning. The method involves setting up and implementation stages. In order to employ KNN, there must exist integer $N > 0$ tuple pairs (x_i, y_i), and the objective is to estimate y_i, in real time, when observing the x valued data set. This data set is pretrained, with labels y, which are assumed to be categorical and binary valued in {0, 1}. The value $K = 1, 2, \ldots, N$ is chosen a priori. It refers to the number of points in the data set used in making a decision on the distance to the query or test value x. The distance function also must be chosen prior to implementation. A norm should be chosen, and the one norm distance will be utilized in the subsequent example.

The method is to calculate the distance from all N points from x_i to x. Then order these distances, along with the corresponding tuple values, in ascending order for distance. Next choose the first K entries. The classification is performed by using the modal value of the labels for the K entries. A few practical aspects of the method follow. Usually K is taken to be an odd integer. Moreover, it is usually much smaller than N. Also several different values for K are often employed, and an optimal K is attempted to be found. Here, the metric might involve the number of distinct entries from those successfully classified, as a function of K. For small problems consisting of two tuples, it is useful to make a visual diagram by placing the unknown test point at the origin of a plot with the elements of the data set in its surroundings. This is illustrated in Fig. 18.3.



Distance	Vector
1/2	(△,1)
1	(□,0)
3/2	(△,1)
√3	(△,1)
2	(□,0)
3	(□,0)

Mode = 1

FIGURE 18.3 K-nearest neighbor classification.

Example 18.6:

Consider the data illustrated in Fig. 18.3. Here there are N = 6 data points: squares with label 0 and triangles with label one. At the origin is the item x, to be classified. The distance, $|x-x_i|$, i = 1, 2, . . ., 6, is calculated and listed from smallest to largest distance, along with their two-tuple vector representation. Since K = 3 in this situation, only the first three entries in the ordering table are examined for the largest number of common labels. Since there are two 1 s and one 0, the mode is 1. So the unknown is classified as a triangle. The error is that among the K = 3 closest distances, only one item was not among the classified entries.#

Using a small value for K makes the KNN technique unstable, that is, it might misclassify. On the other hand, larger values of K make the method more stable due to the majority voting protocol, that is, using the mode. Usually the method is conducted with a relatively small value of K and then gradually increasing the values of K. This increase is halted once the number of items distinct from those classified becomes too large. The method is easily extended for other than binary classification. Additionally, the procedure works for data sets in n dimensions, that is, n + 1 tuples in each vector. The last tuple is again a categorical label. It can be binary, trinary, etc.

18.6 K-nearest neighbor regression

KNN regression is different from KNN classification in numerous ways. First of all, it is employed on vectors (x_i, y_i), for which the labels y_i can take on continuous values, say in R. Additionally, to begin, it is assumed that the data set, $\{x_i\}$, is real valued and monotonic, $x_i < x_{i+1}$, for i = 1, 2, . . ., N. Moreover, for each x_i a positive integer m_i, number of labels, is associated. Thus, there exist (x_i, y_{i_1}), (x_i, y_{i_2}),. . ., $(x_i, y_{i_{m_i}})$ vectors. The actual regression process amounts to forming a function f, defined on the data set $\{x_i\}$. It is such that $f:\{x_i\} \to R$, where the empirical mean estimator is used, $f(\hat{x_i}) = 1/K \sum_{j=1}^{K} y_{i_j}$. Furthermore, the K values y_{i_j} that are employed in the sum are closest to each other using some metric, or clustering algorithm. Alternatively, a nominal, test value y_i is chosen for each x_i, and K values of y_{i_j} are utilized in the average, all of which are closest to y_i.

The value K must be less than m_i. In any case, K always could be equal to one, but in this situation the regression is an overfit. The resulting function would be very good for the trained data, but not so good using different data. In general, there is an abundance of labels, and in this situation, K could be set equal to or less than the minimum of all the m_i. Similar to the classification KNN, different values of K are used to determine the best fitting function.

Example 18.7:

Suppose it is thought that the data set $\{x_i\}$ has labels that approximately follow a square law path, that is, $f(x_i) = y_i = x_i^2$. If a data point say $x_i = 1$ is chosen along with its labels, the vectors are (x_i, y_{i_1}), (x_i, y_{i_2}),. . ., $(x_i, y_{i_{m_i}})$. Assume that these vectors are (1, 1.3), (1, 0.7), (1, 0.3), (1, 1.4), (1, 1.2). Here, $m_i = 5$, and say that K = 3. Then since, $y_i = x_i^2 = 1$, in this

example the distance from 1 to all five labels must be found as in usual KNN methods. Again, only the closest three are chosen. In this case, the three vectors are (1, 1.3), (1, 0.7), and (1, 1.2). It is the labels in the second tuple that are used in forming the average. Thus, $f\hat{}\ (x_i) = 1/K \sum_{j=1}^{K} y_{i_j} = f\hat{}\ (1) = 1/3 \sum_{j=1}^{3} y_{i_j} = 1/3(1.3 + .7 + 1.2) = 1.06.\#$

KNN regression can be performed using n-dimensional data sets, where again it is assumed that there is a strict order in the data, and the labels are from a continuous sort.

18.7 Quantum K-means applications

Several recent articles have appeared involving quantum K-means applications. For instance, Kavitha and Kaulgud (2022) employ unsupervised training methods for data mining and machine learning in determining heart disease using quantum circuits. A brief description is provided in the next paragraph. Quantum techniques have shown to perform better than classical techniques in several areas such as classification, detection, and tacking, as well as in optimization (Shah and Gorard, 2019). A somewhat related work was performed by Singh and Bose (2021); here a quantum optimization scheme was employed in a K-means algorithm for use in CT chest images. A quantum approach for solving mean clustering using QUBO model is explained in Date et al. (2021). Additionally, Khan (2019) used quantum and destructive interference along with rotations to implement K-means clustering. Although Grover's search provides a speedup relative to classical techniques, it is not able to increase the ability to form clusters (Flick et al., 2017).

In the first referenced paper, for quantum K-means clustering, Kavitha and Kaulgud (2022) employed data preprocessing for quantum that involves the PCA along with outlier rejection techniques. Quantum circuits are created for centroid and distance calculation methods. Centroid and cluster updates are performed in an iterative fashion until convergence is achieved. Quantum processing to form clusters is shown to be the paramount improvement area in speed. Prior to performing the K-means clustering, data was converted into quantum states using the method given in Benlamine et al. (2020). The quantum K-means clustering is achieved using three subroutines. First, a swap test gate is employed to calculate the distance between a centroid cluster $|\phi>$ and a data point $|\Psi>$; this is followed by cluster updates and centroid updates.

Also in this paper, a tensor product is given involving the centroid qubit and the data qubit. An ancillary qubit $|0>$ is also involved. These are inputted into the circuit in order $|0\ \Psi\phi>$. A Hadamard gate is applied to the ancillary qubit, thus giving $1/2^{1/2}\ [|0\ \Psi\phi> + |1\ \Psi\phi>]$. This is followed by a controlled swap gate to assign each state to its nearest cluster. In conclusion, comparable results were achieved between classical and quantum means approaches. However, less execution time was experienced for the quantum system.

18.8 Support vector classifiers

Support vector classification (SVC) is a binary classification procedure that can be extended to multiclass classification. In terms of machine language, a training set consists of n pairs of vectors: $\{(v_1, y_1), (v_2, y_2), \ldots, (v_n, y_n)\}$, where v_i are in $H = R^N$ and y_i are in

$\{-1, 1\}$. The objective is to classify the v_i by finding a linear decision hyperplane A, along with a margin. Technically, a margin consists of a pair of hyperplanes each parallel to A and each equidistant from A. These margins are also called gutters or curbs. Additionally, the area between them is thought of as a street, with A as the center line. Points lying on the curb are called support vectors, and this is how SPV machines acquire their name. SPV machines are often called maximum margin classifiers, usually when the data is separable. Another name for this machine is a hard margin classifier. Soft margin classifiers allow for some miss-classification-type errors to occur. This is allowed because they employ cross-validation. This results in better overall classification.

Mathematically, the affine classifying hyperplane is $A = \min(|<w, v_i>+b|)$ for $i = 1, 2, \ldots, n$. Let $y_i = 1$ whenever it is y_i in $A^+ = \{v_i$ in H, such that $<w, v_i>+b$ is greater than or equal to zero$\}$. Likewise, let y_i equal -1, when y_i is in $A^- = \{v_i$ in H, such that $<w, v_i>+b$ is less than zero$\}$. Normalization can be employed in separable circumstances. Assume that the scaling is adjusted and that normalization occurs in the following. Recall that $A = \{v$ in H, such that $<w, v>+b = 0\}$. Now the two gutters, that is, the parallel affine spaces one unit from A, are given by: $A^+ = \{v$ in H, such that $<w, v>+b = 1\}$ and $A^- = \{v$ in H, such that $<w, v>+b = -1\}$. Also, by the scaling, the minimization is $|<w, v_i>+b| = 1$. By requiring the margin to be as large as possible, this can be converted into a maximization problem. First, by noticing that for y_i $(<w, v_i>+b)$, it is greater than or equal to one for all y_i in A^+ as well as in A^-.

Using the aforementioned facts, solving a constrained convex maximization problem will find A as well as the maximum margin. In the following, it is assumed that the data is linearly separable. If this is not the case, the method fails. Moreover, the method is called a hard margin classifier. Finally, the Lagrange multipliers technique will be used to maximize or minimize an objective function with constraints. This will lead to a quadratic optimization that almost always needs an approximate numerical solution. However, a very simple example is provided below, but even here a lot of computation is required.

To begin, finding the max $\|w\|^{-1}$ with constraints: $y_i(<w, v_i> + b)$ is greater than or equal to one, for all y_i. This maximization problem can be replaced by finding the $\min\|w\|^2/2$, with the same constraints. So, the first step is to form the Lagrangian: $L = 1/2 <w, w> - \sum a_i[y_i (<w, v_i>+b) - 1]$. Here, the a_i are called the Lagrange multipliers; they also need to be found. Next, take the partials derivatives or the gradient with respect to w, b, as well as all the a_i, and then set them all equal to zero. Do this first with $\partial L/\partial w = w - \sum a_i\ y_i\ v_i = 0$, and then, $\partial L/\partial b = \sum a_i\ y_i = 0$. Substituting the quantity, $w = \sum a_i\ y_i\ v_i$, back into L gives the long expression $L = 1/2 <\sum a_i\ y_i\ v_i, \sum a_j y_j v_j> - \sum a_i y_i[<\sum a_j y_j v_j, v_i>] - \sum a_i y_i b + \sum a_i$. Next using $\sum a_i y_i = 0$ and simplifying the equation gives $L = -\sum a_i - 1/2 \sum\sum a_i a_j y_j y_i <v_i, v_j>$. Now using the decision rule with w, $<w, v>+b$, that is, its relationship with one, $w = \sum a_i y_i v_i, \sum a_i y_i <v_i, u> + b. L = -\sum a_i - 1/2 \sum\sum a_i a_j y_j y_i (<v_i, v_j>)$.

Example 18.8:

Consider the training set: $\{(v_1, y_1), (v_2, y_2), (v_3, y_3)\}$, where v_i are in $H = R^2$ and y_i are in $\{1, -1\}$. To find the SVC A, where $A = \{v$ in H, such that $<w, v>+b = 0\}$. Above A should be those vectors with $y_i = 1$, and below A should be those vectors such that

$y_i = -1$. Say that the vectors v_i with $y_i = 1$ are $v_1 = (-1\ 0)'$ and $v_2 = (0\ 1)'$, and the one with $y_i = -1$ is $v_3 = (0-1)'$. Form the inner product Gram matrix G, with entries $<v_i, v_j>$, $i = 1, 2, 3$ as row indicators, and $j = 1, 2, 3$ as column indicators. The matrix is given as follows:

$$\begin{matrix} |1 & 0 & 0| \\ |0 & 1 & -1| \\ |0 & -1 & 1| \end{matrix}$$

The Lagrangian for this problem is $L = -\sum a_i - 1/2 \sum\sum a_i a_j y_j y_i (<v_i, v_j>)$. In particular, expanding out provides $L = -(a_1 + a_2 + a_3) - 1/2(a_1^2 + a_2^2 + a_2a_3 + a_3a_2 + a_3^2)$. Here, the inner products $<v_i, v_j>$ are substituted using values from the Gram matrix G. Taking the partial derivatives of L: $\partial L/\partial a_1 = -1 - a_1 = 0$; $\partial L/\partial a_2 = -1 - a_2 - a_3$; $\partial L/\partial a_3 = -1 - a_3 - a_2$. From this, it is seen that $a_1 = -1$. The other two equations are dependent, $a_2 + a_3 = -1$. However, using the constraint, $\sum a_i y_i = 0$, this implies that $a_2 - a_3 = 1$. Consequently, $a_2 = 0$ and $a_3 = -1$. Finally, using $w = \sum a_i\ y_i\ v_i$, it follows that $w = -v_1 + v_3 = (1\ 0)' + (0-1)' = (1-1)'$. Therefore, the affine line classifier happens to be a subspace of R^2, in x-y coordinates; it is $y = x$. The margin here is $1/2^{1/2}$, and support vectors are $v_1 = (-1\ 0)'$ and $v_3 = (0-1)'$.#

When the data is not linearly separable, slack variables are often introduced to make the constraining equations less rigid. The constraints now become as follows: y_i ($<w, v_i> + b$) is greater than or equal to one minus slack values s_i, usually taken to be nonnegative. So again, the object is to minimize min $\|w\|^2/2$, but the s_i cannot be made too large. A modified Lagrangian problem is formulated: min$[\|w\|^2/2 + C/n \sum s_i,]$ where the minimization is for w, b, and s_i. The constant C is a positive parameter, n is the number of data points, and the sum is from 1 to n. The value C balances the importance of regularization and the importance of matching the training set correctly. When C is too large, the training set is matched well; if C is too small, this benefits overfitting. In the latter case, this might be good for training, but maybe not so good for unseen real data. C is called the penalty parameter for miss-classification.

Using the constraining equation gives the formula for L, where a_i and c_i are Lagrange multipliers. $L = \|w\|^2/2 + C/n \sum s_i + \sum a_i (1 - s_i - y_i (<w, v_i> + b)) - \sum s_i c_i$. As before, taking the partial derivatives or gradient of L, with respect to w, b, and s_i, and setting them equal to zero gives $\partial L/\partial w = w - \sum a_i\ y_i v_i = 0$; $\partial L/\partial b = -\sum a_i\ y_i = 0$; $\partial L/\partial s_i = C/n - a_i - c_i = 0$. As in the strict margin classifier, substituting these partials into L winds up with the dual objective function below, and as such it needs to be maximized or its negative minimized. Accordingly, the constraint minimization occurs: min $1/2 \sum\sum a_i a_j y_j y_i <v_i, v_j> - \sum a_j$, with minimization over a_i, and constraints $\sum a_i\ y_i = 0$, and a_i greater than or equal to zero, but less than or equal to C/n. This is a consequence of $C/n - a_i - c_i = 0$ and having c_i greater than or equal to zero.

As mentioned previously, the solution involves a quadratic program. Unlike the aforementioned example, the Gram matrix G will not be used explicitly. Here, an n by n matrix H with entries $H_{ij} = y_j y_i <v, c, v_j>$ will be utilized. Note that if vector matrix notation is employed then the quadratic minimization is min $1/2\ a'\ H\ a - a'\ I$. It is such that $a'\ y = 0$, with 0 less than or equal to a_i less than or equal to C/n. In this case, a is the vector of a_i, y is the vector of y_i, and I is the identity vector.

18.9 Kernel methods

This section is a continuation of Sections 1.10 and 5.7. Recall, in machine learning algorithms, the kernel is employed directly without computing the feature transformation, Φ mapping vectors in the original space. Here, the identity is exploited, $K(v,w) = <\Phi(v), \Phi(w)> = (<v, w>)^d$. An interesting application, not previously illustrated, involves the nonhomogeneous polynomial kernel. Specifically, $K(v,w) = (<v, w>+a)^d$, where a is a real number. By the binomial expansion: $(<v, w> +a)^d = \sum_{k=0}^{d} d!/(k!(d-k)!) <v, w>^k a^{d-k}$. An example illustrating this kernel follows.

Example 18.9:

Let v and w be in R^2 and a in R; then let $A = (<v, w> +a)^2 = (v_1^2 w_1^2 + 2v_1v_2w_1w_2 + w_2^2v_2^2) + 2a(v_1w_1 + w_2v_2) + a^2$. Rearranging: $A = (v_1^2w_1^2 + w_2^2v_2^2) + 2(v_1v_2w_1w_2) + a(v_1w_1 + w_2v_2) + a^2 = \sum_{i=1}^{2} v_i^2w_i^2 + \sum_{i=1}^{2}\sum_{j=1}^{i-1}(2^{1/2}v_iv_j)(2^{1/2}w_iw_j) + \sum_{i=1}^{2}(2a)^{1/2}v_i(2a)^{1/2}w_i + a^2$. This representation is important because it can generalize easily into higher dimensions. Also it will enable Φ to be found. Here, the kernel $= <\Phi(v), \Phi(w)>$. Both arguments within the inner product are the same, so factoring would yield $\Phi(v)$.

Accordingly, the mapping $\Phi: R^2 \to R^6$ is being employed in this example. The feature map Φ employs the two tuples of $v=(v_1v_2)'$, in R^2. It maps them into all possible second, $d = 2$, degree nonhomogeneous monomials, resulting in $\Phi(v)=(v_1^2 v_1^2 2^{1/2}v_1v_2(2a)^{1/2}v_1 (2a)^{1/2}v_2a)'$. Note here that commutativity does not matter. However, by using the same protocol as in the homogeneous case, then the mapping is $\Phi: R^2 \to R^7$, where distinct labeling does matter, that is, even though $v_1v_2 = v_2v_1$, each one occupies a distinct tuple. Thus: $\Phi(v)=(v_1^2v_1^2v_1v_2v_2v_1(2a)^{1/2}v_1(2a)^{1/2}v_2a)'$.#

Recall that a kernel is a function involving a nonempty set X, such that K: X x X → R. It is linked to a map: Φ: X → H, where H is a Hilbert space over the reals. Additionally, for x and y are in X, and K defined by the inner product: $K(x,y) = <\Phi(x), \Phi(y)>$, the function Φ is called the feature map. Also, K is called the feature kernel, and H is sometimes called the feature space. H need not be a RKHS. On the other hand, a kernel in a RKHS is also a feature kernel. Feature kernels are symmetric and positive semidefinite: Let i and j be in any finite set of positive integers, and x_i, x_j in X and a_ia_j real valued, then $\sum_{i,j} a_ia_jK(x_i, x_j) = \sum_i \sum_j a_ia_j <\Phi(x_i), \Phi(x_j)> = \sum_i a_i <\Phi(x_i), \sum_j a_j\Phi(x_j)> = <\sum_i a_i\Phi(x_i), \sum_j a_j\Phi(x_j)> = \| \sum_j a_j\Phi(x_j) \|^2$ which is greater than or equal to zero. In terms of vectors, a′ K a is greater than or equal to zero, where a is a column vector consisting of all the a_i, a′ is a row vector of a_i, and K is a Gram matrix.

Kernels are defined in terms of sequences in l^2. Kernel methods form a firm foundation for machine learning involving support vector-type classification. In this type of classification, hyperplanes are utilized usually in binary categorization. These planes are employed in separating two distinct data classes or objects. Indeed, these planes normally do not pass through the origin and are instances of an affine transformation. Such transformations preserve lines and keep parallels parallel. Additionally, they preserve convexity as well as extreme points. They do not preserve distance or angles. In machine learning, they are also useful because they are an automorphism of affine spaces; they preserve

dimension and subspaces. The data to be categorized into two groups is first assumed to be linearly separable. This preassumes that there exists a hyperplane that can separate the data into two parts. Soft categorization is given for which the data is nonlinear separable. Soft categorization refers to the fact that data is marginally nonseparable, that is, there are occasional outliers. Also, applications of kernels in regression analysis exist. Here, parameter settings are determined that minimize overfitting or under fitting of the data by regressed curves.

Some combinations of feature maps are also feature maps. In particular, the sum and the product of feature maps is a feature map along with nonnegative scalar products. However, the difference of two feature maps might not be a feature map. These properties are used in the construction described later.

Consider a nonempty set X, and a sequence Φ_1, Φ_2, $\Phi_3 \ldots$, of l^2 functions such that Φ_i: $X \to R$. Then $\Phi(x) = (\Phi_1 \Phi_2 \Phi_3 \ldots)$ is the feature map, and $K(x, y) = \sum_{i=1}^{\infty} \Phi_i(x)\Phi_i(y)$ is a kernel on X. This follows using the CBS inequality, that is, $|K(x, y)| = |\sum_{i=1}^{\infty} \Phi_i(x)\Phi_i(y)|$ is less than or equal to $\sum_{i=1}^{\infty} |\Phi_i(x)|^2 \sum_{i=1}^{\infty} |\Phi_i(y)|^2$.

Hyperplane A will be described herein in finite real inner product spaces $H = R^n$. These planes will be an affine subset of H of degree less than n. For b in R, and for w in H, A = {v in H, such that $< w, v > + b = 0$}.

Example 18.10:

Let $H = R^2$, for $b = 2$, and $w = (1\ 4)'$ in H, $A = \{v = (v_1 v_2)'$ in H, such that $< (14)', (v_1 v_2)' > + 2 = 0\}$. Accordingly, A is the straight affine line: $v_1 + 4v_2 = -2$.#

Support vector machine is supervised learning and needs training data. The kernel trick transforms data to a higher dimension, performs classification, and then transforms back. Points in space have tuples that are called features. The hyperplane partitions H into two half planes: $A^+ = \{v$ in $H | < w, v > + b$ greater than or equal to $0\}$ and $A^- = \{v$ in $H | < w, v > + b < 0\}$, thus giving the hyperplane the decision boundary for binary classification. H is a metric space because it is an inner product space and therefore it is also a normed vector space. Indeed, the inner product provides a natural metric d. The distance between v and w in H is $d(v, w) = < v - w, v - w >^{1/2} = \|v - w\|$. The last quantity is the norm induced by the inner product. For a set of vectors, $X = \{v_1, v_2, \ldots, v_n\}$ in H, the distance from a single point v_i in this set to the affine classifier A is given by $| < w, v_i > + b | / \|w\|$. This smallest distance from a vector in X to A is given by minimizing this expression. Multiplying an affine space by constant, that is, multiplying w and b by the same constant, does not change the space A; actually the two spaces are equivalent. Thus the minimum distance can be found by $\min(| < w, v_i > + b|)$ for i = 1, 2, . . ., n. See Section 2.9, involving affine spaces.

Example 18.11:

As in the previous example, a hyperplane will be created. Then the distance will be found from A to the closest vector in the set X. Again, let $H = R^2$, with bias $b = 3$, and $w = (1\ 2)'$ in H, $A = \{v = (v_1 v_2)'$ in H, such that $< (1\ 2)', (v_1 v_2)' > + 3 = 0\}$. Consider the set X = {(0

0)′, (2 3)′, (−2−4)′}. The hyperspace line is $v_1+2v_2=-3$. The distance from (0 0)′ to A is $|<w, (0\ 0)'>+b|/\|w\| = |0+3|/5^{1/2}=3/5^{1/2}$.

The distance from (2 3)′ to A is $|<w, (2\ 3)'>+b|/\|w\| = |8+3|/5^{1/2}=11/5^{1/2}$.

The distance from (−2−4)′ to A is $|<w, (-2-4)'>+b|/\|w\| = |-10+3|/5^{1/2}=7/5^{1/2}$. Classification of all the points in X yields the following: (0 0)′ and (2 3)′ are in A^+; also (−2−4)′ is in A^-. In any case, the zero vector is the closest. Remembering that an affine space can be scaled, and so, multiplying by the absolute norm value of the weight w, the distance formulation is normalized, and the distances become 3, 11, and 5, respectively. By knowing the minimum distance, further scaling could be employed. Indeed, a scaling by $3\|w\|$ would normalize the results.#

18.10 Radial basis function kernel

Radial basis function (RBF) kernel uses, for instance, v = (x y)′ in R^2, and the mapping $\phi(v)$ provides an infinite-dimensional vector $\Phi(v)=(1x\ y\ xy\ yx\ x^2y^2x^2yy^2x\ldots)'$. The kernel method avoids calculating Φ(v), which produces the single, the double, the triple interactions, etc., between the tuples of v. Instead the inner product is performed in R^2, and then the kernel: $k(v,w)=<\Phi(v),\Phi(w)>=e^{-1/2\|v-w\|^2}$. To show formally that this is an inner product kernel, write $e^{-1/2\|v-w\|^2}=e^{-1/2[(v-w)'(v-w)]}=e^{-1/2[v'v-2v'w+w'w]}=e^{-1/2[v'v+w'w]}e^{v'w}$. Let $C=e^{-1/2[v'v+w'w]}$. Then $e^{-1/2\|v-w\|^2}=Ce^{v'w}=Ce^{v'w+1}e^{-1}=C'e^{v'w+1}$ where C′ is a new constant. Thus $e^{-1/2\|v-w\|^2}=\sum_{n=0}^{\infty}(1+v'w\}^n/n!$ This shows that the RBF kernel makes all nonhomogeneous polynomial kernels. This kernel projects into infinite-dimensional space. It performs an infinite number of interactions between data.

The RBF model is founded on using the distance between points in space. If data consists of the pair (x_n, y_n), this influences h(x) based on $\|x-x_n\|$. The standard form is $h(x)=\sum_{n=1}^{N} w_n e^{-\gamma\|x-x_n\|^2}$. The expression is a functional form for a hypothesis. In most cases, the objective is to find w consisting of tuples w_n, n = 1, 2,… N, based on data $D=(x_n, y_n)$. To solve for w: $\sum_{n=1}^{N} w_n e^{-\gamma||x-x_n||^2}=y_m$ for $x=x_m'$. So $\Phi\ w=y$. The solution is $w=\Phi^{-1}y$, similar to regression. For classification, use $h(x)=\text{sgn}(\sum_{n=1}^{N} w_n e^{-\gamma||x-x_n||^2})$. Let $s=\sum_{n=1}^{N} w_n e^{-\gamma||x-x_n||^2}$.

18.11 Bound matrices

Similar to time-limited signals described in Section 1.9, two-dimensional time-limited matrices form an inner product space (Alotto et al., 1998; Dougherty and Giardina, 1987). These are real-valued functions f with finite support in R^{ZxZ}. This is the carrier set corresponding to VECTOR, and for SCALAR, the reals are employed. The function f in R^{ZxZ} means that f: Z x Z→R, but all these functions described by bound matrices have nonzero values on a finite subset of lattice points. They will be denoted by A. Note that V-ZERO corresponds to the 0 entity in A. Signals or functions in A form a real-valued vector space using point-wise operations. As in the referenced section, A also forms an inner product space, using the dot product on the intersection of these matrices, but again, it is not a

Hilbert space. Additionally, the cozero set, complement of zero, for f in A, is denoted by COZ(f). This finite set consists of all lattice points, (n, m) in Z x Z where f (n, m) differs from V-ZERO.

A convenient representation for any signal f in A, such that COZ(f) is not empty, is described similar to the Young diagrams in Section 15.7. Indeed, a critical lattice point (p, q) is the location of the leftmost, uppermost element of any matrix enclosing all nonzero values of f. For instance, find the point (p, q) equaling the smallest value n, for which f (n, $\cdot$) that is nonzero is located, and also the largest value of m such that f ($\cdot$, m) that is nonzero is located. Next, a matrix-type structure is employed to enclose all the nonzero values of f, along with (p, q). The latter point itself might be zero. This is similar to a minimal enclosing rectangle, although the matrix need not be minimal. However, in general, there has to be at least one nonzero value in the matrix and no nonzero values outside. Once the leftmost coordinate p and uppermost value coordinate q are located, the value (p, q) is used as a pointer on the lower right-hand side of the matrix. This data structure is called a bound matrix. Throughout, it is at least initially assumed that the cozero set is nonempty. An example might be helpful.

Example 18.12:

Consider the two-dimensional digital signal f in A, where f is zero except on COZ(f) = {(0, 0), (−1, 1), (1, 2), (1, 0), (1, 1)}. Also say that f (0, 0) = 5, f (−1, 1) = 3, f (1, 2) = 2, f (1, 0) = −3, and finally, f (1, 1) = 6. The first x coordinate set of COZ(f) consists of points in Z, which is {0, −1, 1}, and the minimum is p = −1. The second y coordinate set is {0, 1, 2}; the maximum is q = 2. Accordingly, two bound matrix structures representing the function f are given by:

$$\begin{matrix} |0 & 0 & 2| \\ |3 & 0 & 6| \\ |0 & 5 & -3|_{-1,2} \end{matrix} \qquad \begin{matrix} |0 & 0 & 0 & 0 & 2 & 0| \\ |0 & 0 & 3 & 0 & 6 & 0| \\ |0 & 0 & 0 & 5 & -3 & 0| \\ |0 & 0 & 0 & 0 & 0 & 0|_{-3,2}. \# \end{matrix}$$

Thus it is seen that there exists an equivalence class of bound matrices representing a function in A, but this direction will not be pursued. However, in the aforementioned example, a form of compression is illustrated. The bound matrix to the left is called a minimal bound matrix, since no smaller matrix than this three-by-three matrix can capture all the nonzero values of f.

The inner product space A also forms an algebra. For f and g in A, the point-wise multiplication is the convolution defined by

BINE(f, g)(n,m) = (f ★ g)(n,m) = $\sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(n\text{-}k, m\text{-}j) f(k, j)$. With this type of multiplication BINE, A becomes a unital, commutative, associative algebra. The unital function V-ONE is $I=(1)_{(0,0)}$, and I ★ f = f ★ I = f. Although the limits in the aforementioned summation are infinite, there are only a finite number of nonzero terms. A more concise expression for the actual summation limits involves the cozero sets. In particular, COZ(f ★ g) is a subset of the dilation of the two sets COZ(f) and COZ(g). The dilation is given by D(COZ (f),COZ(g)). This quantity equals the union of all sets of integer pairs {(n + k, m+j)}, where (n, m) is in COZ(f) and (k, j) is in COZ(g). The convolution of f and g will be zero outside

this dilated set. Thus the dilation provides a support region for the convolution. Therefore, the convolution can be calculated using

$$(f \bigstar g)(n, m) = \sum f(n - k, m - j)g(k, j),$$

where (k, j) is in COZ(g) and $(n - k, m - j)$ is in COZ(f). The procedure for finding $(f \bigstar g)$ (n,m) is to first find the dilated set D(COZ(f),COZ(g)). Only if (n,m) is in this set, then use (k, j) in COZ(g); if $(n - k, m - j)$ is in COZ(f), then multiply $f(n - k, m - j)$ by g(k,j). This must be calculated for all possibilities, and the sum should then be taken. This result is the convolution at (n, m). Because this sum might equal zero, it shows that the dilated set might not be the cozero set for the convolution.

Example 18.13:

Consider the two bound matrices f and g, in A, given by minimal bound matrix structures, respectively:

$$\begin{matrix} |0 & 1| & |2 & 0| \\ |2 & 3|_{0,2} & |-1 & 4|_{1,3} \end{matrix}$$

COZ(f) = {(1, 2), (0, 1), (1, 1)} and COZ(g) = {(1, 3), (1, 2), (2, 2)}. The dilation of these two sets is D(COZ(f),COZ(g)) = {(2, 5), (2, 4), (3, 4), (1, 4), (1, 3), (2, 3), (3, 3)}. This set of lattice points will only provide a support region for the convolution. Outside this region, all zeros appear at lattice points. Beginning with the first element in this set, (n, m) = (2, 5), use (k, j) = (1, 3) in COZ(g), and then see if $(n - k, m - j) = (1, 2)$ is in COZ(f). Yes, it is, so find the product, $f(1, 2) \cdot g(1, 3) = 1 \cdot 2 = 2$. Next, the second element in this set (n, m) = (2, 4), use (k, j) = (1, 3) in COZ(g), and then see if $(n - k, m - j) = (1, 1)$ is in COZ(f). Yes, it is, so find the product, $f(1, 1) \cdot g(1, 3) = 3 \cdot 2 = 6$. But (k, j) = (1, 2) also works, that is, $(n - k, m - j) = (1, 2)$ is in COZ(f); accordingly $f(1, 2) \cdot g(1, 2) = 1 \cdot (-1) = -1$. These must be added together; thus $(f \bigstar g)(2, 4) = 6 - 1 = 5$.

Next, the third element in this set (n, m) = (3, 4), use (k, j) = (1, 3) in COZ(g), and then see if $(n - k, m - j) = (2, 1)$ is in COZ(f). No, then try (k, j) = (1, 2) in COZ(g); then see if $(n - k, m - j) = (2, 2)$ is in COZ(f). Again no, then try (k, j) = (2, 2) in COZ(g); then see if $(n - k, m - j) = (1, 2)$ is in COZ(f). Yes, it is, so find the product, $f(1, 2) \cdot g(2, 2) = 1 \cdot 4 = 4$. Next, the fourth element in this set (n, m) = (1, 4), use (k, j) = (1, 3) in COZ(g), and then see if $(n - k, m - j) = (0, 1)$ is in COZ(f). Yes, it is, so find the product, $f(0, 1) \cdot g(1, 3) = 2 \cdot 2 = 4$. Next, the fifth element in this set (n, m) = (1, 3), use (k, j) = (1, 2) in COZ(g), and then see if $(n - k, m - j) = (0, 1)$ is in COZ(f). Yes, it is, so find the product, $f(0, 1) \cdot g(1, 2) = 2 \cdot (-1) = -2$.

Next for the sixth element in the dilated set, (n, m) = (2, 3), use (k, j) = (1, 2) in COZ(g), and then see if $(n - k, m - j) = (1, 1)$ is in COZ(f). Yes, it is, so find the product, $f(1, 1) \cdot g(1, 2) = 3 \cdot -1 = -3$. However, (k, j) = (2, 2) also works since if $(n - k, m - j) = (0, 1)$ is in COZ(f). Then $f(0, 1) \cdot g(2, 2) = 2 \cdot 4 = 8$. Adding yields $-3 + 8 = 5$. Finally, the last value in the dilated set is (n, m) = (3, 3), and (k, j) = (2, 2) works because $(n - k, m - j) = (1, 1)$ is in COZ(f). Therefore, $f(1, 1) \cdot g(2, 2) = 3 \cdot 4 = 12$.

Consequently, the bound matrix for $f \bigstar g$ is the following:

$$\begin{matrix} |0 & 2 & 0| \\ |4 & 5 & 4| \\ |-2 & 5 & 12|_{1,5} \end{matrix} .\#$$

As in the bound vector description in Section 3.2, there exists a parallel algorithm for finding bound matrix convolution. It too is illustrated in Figure 3.2. In short the method is

$f \bigstar g = ADD[SCALAR(g(n, m), TRAN(f; n, m)), for(n, m)inCOZ(g)$.

SCALAR is just multiplication of translates of the matrix for f, by values of g, located at (n, m). TRAN just translates the matrix for f by changing the location pointer on the lower right outside corner of the matrix. The change occurs by adding, resulting in (p+n, q+m). The values within the matrix do not change by only using this operation. The difference between the parallel convolution equation mentioned earlier and the referenced figure is none. SCALAR and TRAN operations commute!

Example 18.14:

Referring to Example 8.13 where $COZ(g) = \{(1, 3), (1, 2), (2, 2)\}$, also $g(1, 3) = 2$, $g(1, 2) = -1$, $g(2, 2) = 4$. Then f is given below, followed by $g(1, 3) \cdot TRAN(f; 1, 3)$, followed by an application of $g(1, 2) \cdot TRAN(f; 1, 2)$, finally multiplying $g(2, 2) \cdot TRAN(2, 2)$:

$$\begin{matrix} |0 & 1| & |0 & 2| & |0 & -1| & |0 & 4| \\ |2 & 3|_{0,2} & |4 & 6|_{1,5} & |-2 & -3|_{1,4} & |8 & 12|_{2,4}. \end{matrix}$$

Rewriting the last three bound matrices with a common location indicator gives the following:

$$\begin{matrix} |0 & 2 & 0| & |0 & 0 & 0| & |0 & 0 & 0| \\ |4 & 6 & 0| & |0 & -1 & 0| & |0 & 0 & 4| \\ |0 & 0 & 0|_{1,5} & |-2 & -3 & 0|_{1,5} & |0 & 8 & 12|_{1,5}. \end{matrix}$$

Adding the aforementioned three bound matrices together gives exactly the same answer as previously given in Example 18.13.#

A bound matrix with minimal support for $f \bigstar g$, where f and g are in A, can be found. Assume that the convolution is not equal to zero. First find D(COZ(f),COZ(g)). Then in this set find the minimum value of the first tuple and the maximum value of the second tuple; call this (p, q). Use this as the location of the left uppermost entry in the bound matrix for the convolution. The actual bound matrix of support is N by M where $N = p' - p, +1$, where p' is the largest value of the first tuple in the dilated set, and $M = q - q' + 1$, where q' is the minimum value of the second tuple in the dilated set.

Example 18.15:

Again refer to Example 18.13, where $D(COZ(f),COZ(g)) = \{(2, 5), (2, 4), (3, 4), (1, 4), (1, 3), (2, 3), (3, 3)\}$. Then $(p, q) = (1, 5)$, and since $p' = 3$ and $q' = 3$, then $N = 3 - 1 + 1 = 3$, $M = 5 - 3 + 1 = 3$.#

Similar to Section 3.9, where wraparound bound vectors were described, wraparound bound matrices also form a Banach* algebra. In this case, functions f in $R^{Z_r x Z_r}$ are utilized. These are functions with domain {0, 1, ... r − 1} x {0, 1, ... r − 1}, with codomain the real numbers. Moreover, Z_r has a cyclic group structure, modular r. The set of all these functions for fixed positive integer r will be denoted by A_r. Bound matrices provide a convenient representation for elements in A_r. Point-wise addition, scalar multiplication, and so on show that this structure becomes a real-valued vector space with V-ZERO being the bound matrix of all zeros. Additionally, convolutional multiplication gives a structure of a unital, abelian, and associative algebra. The unity element, V-ONE, is a bound matrix of all zeros, except that at the (0, 0) location there is a one. Finally, a Hilbert space structure is created using the l^2 norm. The completeness follows as in Appendix A.1.

Just like the one-dimensional case of wraparound bound vectors, a superscript (W_r) will appear on the upper right-hand corner for modulo r bound matrices. It also indicates an r^2 size of bound matrix is being utilized. Additionally, a subscript outside the lower right-hand side indicates the upper left-hand entry of the wraparound bound matrix. An example should help make some of these concepts more understandable.

Example 18.16:

Consider the wraparound bound matrix, f in A_3; it is given below as described earlier. This is followed below by the scalar multiple of f by 2, that is, SCALAR(f; 2). Finally, the third matrix below is a more compact representation for this actually three-by-three matrix:#

$$\begin{matrix} |0 & 2 & -1|^{W3} \\ |0 & 0 & -3| \\ |0 & 0 & 0|_{0,3} \end{matrix} \quad \begin{matrix} |0 & 4 & -2|^{W3} \\ |0 & 0 & -6| \\ |0 & 0 & 0|_{0,2} \end{matrix} \quad \begin{matrix} |2 & -1|^{W3} \\ |0 & -3|_{1,3} \end{matrix}$$

Additionally, the two norm of f is $\|f\|_2 = (4+1+9)^{1/2} = 14^{1/2}$, whereas the one norm is $\|f\|_1 = 6$.#

The cozero set for f in A_r consists of {(n, m) in $Z_r x Z_r$ such that f (n, m) in R−{0}}. As before, the dilated set for two functions in A_r is of the uttermost importance in calculating the convolution. The dilated set is a support region for the convolution, and consequently only points within this set should be employed for this operation. The convolution can be calculated using:

$$(f \bigstar g)(n, m) = \sum f(n - k, m - j)g(k, j),$$

where (k, j) is in COZ(g) and (n − k, m − j) is in COZ(f). The procedure for finding (f ★ g) (n,m) is, as before, to first find the dilated set D(COZ(f),COZ(g)). Only if (n,m) is in this set, then use (k, j) in COZ(g); if (n − k, m − j) is in COZ(f), then multiply f(n − k, m − j) by g(k,j). This must be calculated for all possibilities, and the sum should then be taken. This result is the convolution at (n, m). Before an example is provided, it is best to introduce the parallel algorithm.

The parallel algorithm for finding wraparound bound matrix convolution is illustrated in Figure 3.2. As before, the method is:

f ★ g = ADD[SCALAR(g(n, m), TRAN(f; n, m)), for(n, m)inCOZ(g).

Again, SCALAR, in this formula, is multiplication of translates of the matrix for f. The multiplication factor is the value of g, located at (n, m). This is similar to the multiplication in the bound matrix situation. TRAN just translates the matrix for f by changing the location pointer on the lower right outside corner of the matrix, but this is performed modulo r. The change occurs by adding. Resulting in (p+n, q+m), all operations modulo r. The values within the matrix do not change with only this operation.

Example 18.17:

Again consider the wraparound function f in A_3 from the previous example. It is given again below followed by g also in A_3. The objective is to find f★g. The operation will be performed using the parallel algorithm, and then a value or two will be checked using the point-wise calculation.

$$\begin{matrix} |2 & -1|^{W3} & |1 & 4|^{W3} \\ |0 & -3|_{1,2} & |0 & 5|_{1,2}. \end{matrix}$$

First finding the cozero sets, COZ (f) = {(1, 2), (2, 2), (2, 1)}, and COZ (g) = {(1, 2), (2, 2), (2, 1)}. The dilation is D(COZ (f), COZ (g)) = {(2, 4), (3, 4), (3, 3), (4, 4), (4, 3), (4, 2)} = {(2, 1), (0, 1), (0, 0), (1, 1), (1, 0), (1, 2)}. The last set of integer pairs occurs since all domain operations are performed modulo three. Reading off the values for matrix g gives g (1, 2) = 1, g (2, 2) = 4, g (2, 1) = 5. Preparing for the parallel algorithm, first f is given below. This is followed by g (1, 2) • TRAN(f; 1, 2). Next, it follows g(2, 2) • TRAN (f; 2, 2). Finally, use g (2, 1) • TRAN (f; 2, 2, 1).

$$\begin{matrix} |2 & -1|^{W3} & |2 & -1|^{W3} & |8 & -4|^{W3} & |10 & -5|^{W3} \\ |0 & -3|_{1,2} & |0 & -3|_{2,1} & |0 & -12|_{0,1} & |0 & -15|_{0,0}. \end{matrix}$$

Rewriting the last three matrices with a common pointer gives:

$$\begin{matrix} |0 & 0 & 0|^{W3} & |0 & 0 & 0|^{W3} & |0 & -15 & 0|^{W3} \\ |-1 & 0 & 2| & |8 & -4 & 0| & |0 & 0 & 0| \\ |-3 & 0 & 0|_{0,2} & |0 & -12 & 0|_{0,2} & |10 & -5 & 0|_{0,2}. \end{matrix}$$

Adding these is simple compared with the earlier, more compact representation. As a consequence, the result is f★g. =

$$\begin{matrix} |0 & -15 & 0|^{W3} \\ |7 & -4 & 2| \\ |7 & -17 & 0|_{0,2}. \end{matrix}$$

As a check, consider f★g (0, 0) = f (1, 2) • g (2, 1)+f (2, 1) • g (1, 2) = 2 • 5+(−3) • 1 = 7. Also f★g (0, 1) = f (1, 2) • g (2, 2)+f (2, 2) • g (1, 2) = 2 • 4+(−1) • 1 = 7.#

18.12 Convolutional neural networks and quantum convolutional neural networks

The principal operation in a convolutional neural network (CNN) is convolution. This operation is often performed at several layers in a CNN architecture. Fig. 18.4 provides a

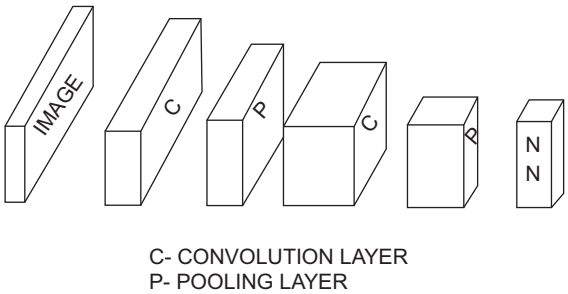


FIGURE 18.4 Convolutional neural network.

typical illustration of the operations within a CNN used in imaging. Very often, in between convolution operations, some type of compression is performed. These operations are illustrated in the figure and are called pooling layers. In these operations, a small number of pixels, usually in a rectangular pattern, are converted into to a single scalar value. This is often done by max pooling, wherein the largest value among all the pixel values is chosen as the representative. Sometimes, the average or median value is calculated and employed. In any case, a source coding has occurred, and the result is a reduced computational burden. The final operation is a typical fully connected NN layer. Often backpropagation is performed to adjust critical parameters involving the convolutional kernels. Convolutional kernels in the past were often handcrafted. However, with the back end NN being utilized often, more useful kernels can be machine learned.

In the referenced figure, C represents a convolutional layer. The leftmost convolutional layer, for instance, might seek out edges or low-level characterization of texture. Adjacent to C is P representing a pooling layer. This layer might reduce the noise level. The next convolutional layer might be used for recognizing more complex structures. The last layer is often a fully connected NN.

In CNN, there exists an observed image f, for instance, consisting of N by M pixels represented by a bound matrix. The convolution process involves a heuristically determined smaller n by m bound matrix g, called the kernel that is convolved with f. The observed image in this case is said to be filtered by g. Usually, the filtering matrix is required to fit into the interior of the observed image. Because of this arrangement, the resulting convolution is of dimensions $N-n+1$ by $M-m+1$. To prevent shrinkage, often the observed image is enlarged by adding a boarder.

Recently, there has been a surge of papers and articles on quantum versions of CNN, QCNN (Hur et al., 2022). They are used in classical machine learning classification contexts and exploit entanglement. QCNNs are also employed in conventional image processing applications such as spatial filtering, edge detection, as well as handwritten symbol recognition (Wei et al., 2022). As in the past, research continues using machine learning devices to classify high-energy events in physics. An instance of this research is the article Chen et al., (2022). Another instance of this type of application can be found in Hermann et al. (2022). In this report, QCNNs are utilized on a superconducting quantum processor for recognizing quantum phases. Some papers explain how QCNNs can be implemented using quantum circuits (Zheng et al., 2022). In this article, a universal QCNN model is proposed consisting of all three: the convolutional layer, the pooling layer, and a fully connected NN layer. Several papers have emerged on medical-type applications using QCNNs (Magallanes et al., 2022).

References

Alotto, L., Giardina, C., Luo, H., 1998. A unified signal algebra approach to two dimensional signal processing. Pure Appl. Math 210 Marcel Dekker.
Benlamine, et al., 2020. Quantum collaborative k means. IJCNN IEEE.
Chen, S., et al., 2022. QCNN for high energy physics data analysis. Phys. Rev. Res.
Date, P., Arthur, D., Pusey-Nazzaro, L., 2021. QUBO formulations for training machine learning models. Sci. Rep. 11 (1), 10029.
Dougherty, E., Giardina, C., 1987. 0-13-453283-X, Image Processing Continuous to Discrete, 1. Prentice-Hall.
Flick, J., Ruggenthaler, M., Appel, H., Rubio, A., 2017. Atoms and molecules in cavities, from weak to strong coupling in quantum-electrodynamics (QED) chemistry. Proc. Natl. Acad. Sci. 114 (12), 3026–3030.
Hermann, J., et al., 2022. Realizing QCNN on a superconducting quantum processor to recognize quantum phases. Nature.
Hur, T., et al., 2022. QCNN for Classical Data Classification. Springer Link.
IonQ Staff, 2023. Picturing the Future With Quantum-Enabled Road Sign.
Kavitha, S.S., Kaulgud, N., 2022. Quantum machine learning for support vector machine classification. Evol. Intell. 1–10.
Khan, H. 2019. Quantum Fluctuations Across the Superconductor-Insulator Transition. The Ohio State University.
Liu, N., Rebentrost, P., 2018. Quantum machine learning for quantum anomaly detection. Phys. Rev. A. 97 (4), 042315.
Magallanes, E., et al., 2022. Expert systems with applications Hybrid Classical QCNN for Stenosis Detection in X-ray Coronary Angiography. Elsevier.
Mengoni, R., 2021. Facial expression recognition on a quantum computer. Quantum Mach. Intell. article 3 no. 8.
O'Malley, D., et al., 2018. Nonnegative/Binary matrix factorization with a D-Wave quantum annealer. PLoS One 13 (12)e0206653. Available from: https://doi.org/10.1371/journal.pone.0206653.
Perrier, et al., 2022. QDataSet, quantum datasets for machine learning. Sci. Data.
Shah, R., Gorard, J., 2019. Quantum cellular automata, black hole thermodynamics, and the laws of quantum complexity. arXiv preprint arXiv:1910.00578.
Singh, P., Bose, S.S., 2021. A quantum-clustering optimization method for COVID-19 CT scan image segmentation. Expert Syst. Appl. 185, 115637.
Valiant, L., 1984. A theory of the learnable. Commun. ACM 27 (11).
Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer.
Vapnik, V., Chervonenkis, A., 1971. On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab. Appl. 16 (2).
Wei, S., et al., 2022. A QCNN on NISQ Devices. Springer Link.
Wittek, P., 2014. Quantum Machine Learning. Google Books.
Yang, C., et al., 2021. Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. ICASS.
Zheng, J., et al., 2022. Design of a QCNN on quantum circuitsElsevier J. Frankl. Inst.

CHAPTER

19

Reproducing kernel and other Hilbert spaces

19.1 Algebraic solution to harmonic oscillator

In Section 6.1, the ground-state solution to the harmonic oscillator is shown to be equal to $\Psi_0 = (\mathrm{m}\ \mathrm{w}_0/(\pi \mathrm{h}))^{1/4} \mathrm{e}^{-m w_0 x^2/(2h)}$. The solution was obtained by solving a first-order differential equation and normalizing a Gaussian-type kernel. The normalization involved finding $\mathrm{N} = (\mathrm{m}\ \mathrm{w}_0/(\pi\ \mathrm{h}))^{1/4}$. For the general quantum harmonic oscillator, the time-independent Schrodinger equation is as follows: $-\mathrm{h}^2/(2\mathrm{m})\partial^2\Psi(\mathrm{x})/\partial \mathrm{x}\partial \mathrm{x} + 1/2\ \mathrm{m}\ \mathrm{w}_0^2\mathrm{x}^2\Psi(\mathrm{x}) = E\Psi(\mathrm{x})$. The objective is to find the allowed energy states E, as eigenvalues and their corresponding eigenfunctions, Ψ (x). These functions should be normalized and either symmetric or antisymmetric about $\mathrm{x} = 0$. Moreover, the probability density function, $|\Psi(\mathrm{x})|^2$, must be finite over the interval $(-\infty, \infty)$.

Following the excellent presentation provided in Hall (2013), the solutions for higher energy, that is, the solutions for excited state levels, will be found in a purely algebraic manner. In the referenced text, one can find important domain issues concerning the unbounded creation and annihilation operators used in this development. From an algebraic viewpoint, the key identities involve the ground state, Ψ_0. First, using the annihilation operation a yields a $|\Psi_0> = 0$. Second, there exist cyclic properties for the creation operation $\mathrm{a}^\dagger{:}(\mathrm{a}^\dagger)^n|\Psi_0> = \Psi_n$, where n is a nonnegative integer. See Section 15.2 for details on the cyclic property of $\mathrm{a}^\dagger$. A result of the cyclic property is that the excited state $\Psi_n = \mathrm{H}_n\Psi_0$. Here, H_n is a polynomial of degree n, and it can be found using induction, starting with $\mathrm{H}_0 = 1$. Additionally, these polynomials are called the Hermite polynomials. They are orthogonal on $(-\infty, \infty)$, a consequence of being solutions to a Sturm-Liouville differential equation. See Appendix A.7. Using the Rodrigues formula for Hermite polynomials: $\mathrm{H}_n = (-1)^n \mathrm{e}^{x^2} \mathrm{d}^n \mathrm{e}^{-x^2}/\mathrm{dx}^n$, n = 0, 1, 2, ... gives, for instance, $\mathrm{H}_0 = 1$, $\mathrm{H}_1 = 2\ \mathrm{x}$, $\mathrm{H}_2 = 4\ \mathrm{x}^2 - 2$, and $\mathrm{H}_3 = 8\ \mathrm{x}^3 - 12\ \mathrm{x}$, ...

Accordingly, the wave function solutions are $\Psi_n = \mathrm{N}\ \mathrm{H}_n(\mathrm{b}\ \mathrm{x})\mathrm{e}^{-(bx)^2/2}$, where $\mathrm{b} = [(\mathrm{mw}_0)/\mathrm{h}]^{1/2}$. These solutions have corresponding eigenvalues given by allowable discrete energy levels $\mathrm{E}_n = (2\mathrm{n}+1)\mathrm{h}\ \mathrm{w}_0/2, \mathrm{n} = 0, 1, 2, \ldots$. Fig. 19.1 illustrates the even symmetry for even-numbered wave functions and odd symmetry for the odd-numbered wave functions. Moreover, when the horizontal coordinate is zero, there is either an extreme

Many-Sorted Algebras for Deep Learning and Quantum Technology
DOI: https://doi.org/10.1016/B978-0-443-13697-9.00027-8

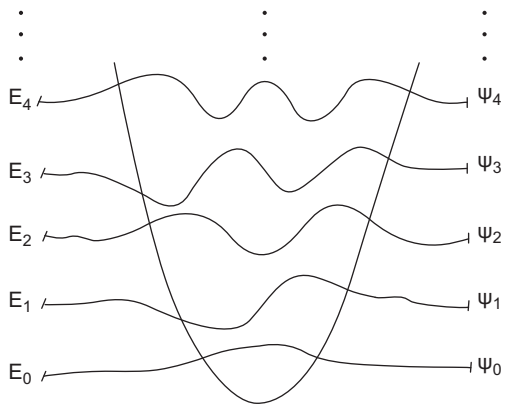


FIGURE 19.1 The first five wave functions for the harmonic oscillator.

point or a zero crossing for each wave function. The zero crossings occur mod2, starting with Ψ_0, and local maxima occur mod4, and so do the local minima.

19.2 Reproducing kernel Hilbert space over C and the disk algebra

To begin, a useful and intuitive definition of H being a reproducing kernel Hilbert space (RKHS) over X means that H is a Hilbert space of complex-valued continuous functions over the nonempty set X. So, H is a subset of C^X. Also, for all functions or vectors, f and g in H, and $\epsilon > 0$, there exists a $\delta > 0$, such that if $\|f-g\| < \delta$, then $|f(x)-g(x)| < \epsilon$. Accordingly, this means that the functions are point-wise continuous. Using RKHS, boundedness and continuity are not related. To see that this is a subtle requirement, consider the following illustration.

Example 19.1:

Let H be the Hilbert space of complex-valued absolute squared functions, on [0, 1]. Consider the two functions $f = t$ on this closed interval and $g = t$ on [0, 1) and $g(1) = 4 + 3i$. In this case, $\|f-g\| = 0$; however, sup $|f-g| = 5$. These functions are bounded, but not continuous.#

In the previous example, it was shown that boundedness in the Hilbert space plays no role in point-wise calculations as far as continuity is concerned. A function K: $X^2 \to C$ is called a reproducing kernel of H, whenever H contains all unary functions for x in X, where $K_x(y) = K(x, y)$, and for every x in X and f in H, then $f(x) = \langle f, K_x \rangle$. The last property is the actual reproducing property.

The key to a rigorous approach for creating a RKHS and relating it to the reproducing property is the Riesz representation theorem (RRT): For a Hilbert space H, with inner product $\langle \cdot, \cdot \rangle$: H x H→C and any continuous linear functional ϕ in H*, there exists a unique vector f_ϕ in H such that $\phi(v) = \langle v, f_\phi \rangle$, for all v in H. Additionally, the norms are equal, that is, $\|f_\phi\| = \|\phi\|$. The first norm is in H, and the second norm is in H*, where- $\|\phi\| = \inf \{C > 0$ such that $|\phi(v)| < C\|v\|$ for all v in H$\}$.

Getting back to the RKHS, there is a linear evaluation functional E_x:H→C (or R in the next section), such that $E_x(f) = f(x)$ is bounded. By the RRT, every bounded linear functional is given by an inner product with a unique vector in H; thus for every x in X, there is a unique function K_x in H such that for all g in H, $g(x) = < g, K_x >$. Following Putnam (2019), K_x is called the reproducing kernel at the point x. The reproducing kernel for H is K: X x X→H, and it is defined by $K(x, y) = < k_y, k_x >$. Also, $|E_x|^2 = < k_x, k_x > = \|k_x\|^2 = K(x, x)$. So a reproducing Hilbert space H, over C, contains a reproducing kernel K, having the reproducing property, $<K(., x), f> = f(x)$, for all x in X and f in H. Additionally, if there exists a reproducing kernel, then H is a RKHS. That is, the linear evaluation function E_x is bounded. Here, $E_x(f) = < K(., x), f > = f(x)$, and so by the CBS inequality, it follows that $|E_x(f)|$ is less than or equal to $\|K(., x)\| \|f\| = < K(., x), K(., x > ^{1/2} \|f\| = K(x, x)^{1/2} \|f\|$, and so $E_x(f)$ is bounded and $|E_x|$ is less than or equal to $K(x, x)^{1/2}$.

The kernel is self-reproducing and positive semidefinite. Use $<K(., x), K(., y)> = K(y, x) = <K(., y), K(., x)>$*. To see that the property, positive semidefiniteness holds, let i and j be in any finite set of positive integers. Then $\sum_{i,j} a_i^* a_j K(x_i, x_j) = \sum_i a_i^* < K(., x_i), \sum_j a_j K(., x_j) > = < \sum a_i^* K(., x_i), \sum_j a_j K(., x_j) > = \| \sum_j a_j K(., x_j) \|^2$ which is greater than or equal to zero. This holds for all a_i in C and x_i in X. Kernels having these properties are sometimes called Mercer kernels. There is a converse to this result. It says whenever a symmetric positive semidefinite kernel K on a set X exists, then there is a unique Hilbert space H for which K produces a RKHS. This theorem is due to Moore-Aronszajn (Aronszajn, 1950). An outline of the results from this paper follows. For any x in X, let K(x,.) be an operator K_x. Define the set of all K_x as a, the linear span for a vector space comprised of all linear combinations of such operators over C. An inner product can be defined for two such linear combinations as $< \sum_{i=1}^{n} a_i K_{xi}, \sum_{j=1}^{m} a_j K_{yj} > = \sum_{i=1}^{n} \sum_{j=1}^{m} a_j a_i^* K(x_i, x_j)$. The properties of an inner product hold since the kernel is symmetric and positive definite. Let H be the completion of this inner product space. This shows that H consists of functions $f(x) = \sum_{i=1}^{n} a_i K_{xi}(x)$ such that the limit supremum as n goes to infinity and positive nonzero m> n, $\| \sum_{i=n}^{n+m} a_i K_{xi} \| = 0$.

To see that the reproducing property holds, $<f, K_x> = < \sum_{i=1}^{n} a_i K_{xi}(x), K_x > = \sum_{i=1}^{n} a_i < K_{xi}, K_x > = \sum_{i=1}^{n} a_i K(x_{xi}, x) = f(x)$. In order to determine that the Hilbert space is unique, let H! be another Hilbert space of functions having K as a reproducing kernel; then for every x and y in X, this other Hilbert space would have inner product $<K_{xi}, K_x>$, which again is $K(x_{xi}, x)$. Using linearity, both inner products are the same on the span using all K_x. Also assume that the completion is performed as before, but for H!. This shows that H is a closed subset of H!. To prove that the inclusion goes the other way, use f in H!, and write $f = f_H + f_H^{\perp}$, where f_H is in H and $f_H^{\perp}$ is in $H^{\perp}$. Because K is the reproducing kernel for both H and H!, it follows using inner products in H! that $f(x) = < K_x, f > = < K_x, f_H > + < K_x, f_H^{\perp} > = < K_x, f_H >$. Because, $< K_x, f_H^{\perp} > = 0$ and since K_x is in H, it follows that the inner product of K_x with $f_H^{\perp}$ in H! is zero. Therefore, the inner product in H! equals the inner product in H, showing that $< K_x, f_H > = f_H$, and so H = H!.

If H is the space of functions from X to C, then a functional on this space can be found for x in X; write $E_x f = f(x)$. Given a set X, a RKHS H is the space of functions from X to C; the evaluation functionals for each x in X, E_x: H→C, are bounded, as originally stated.

Example 19.2:

The Hardy space is a Hilbert space, H^∞; it consists of all analytic functions f, on the open unit disk, D in C. The mean square value of these functions on any circle of radius r is bounded as r approaches one from the interior. That is, $\|f\|^2 = \sup[1/(2\pi) \int_0^{2\pi} |f(re^{i\theta}|^2 d\theta] < \infty$, where the supremum is taken over all $0 < r < 1$ (Katznelson, 1976). As a result of Fatou's theorem (Krantz, 2007), the analytic function can be extended to a function in L^2 on the boundary of the disk. It becomes the point-wise limit almost everywhere. The limit is always taken from the interior and never a tangential limit. Also, using the uniform norm for f, that is, $\|f\| = \max \{f(z) \text{ such that } z \text{ is in clos}(D)\}$, H^∞ is a commutative Banach algebra. For f(z) in H^∞, $f(z) = \sum_{n=0}^{\infty} a_n z^n$ analytic in unit disk D, and such that $\sum_{n=0}^{\infty} |a_n|^2 < \infty$. This is called the disk algebra.

To construct the Hardy RKHS, let L: $H^\infty \to l^2$, where $L(f) = (a_0\ a_1 \ldots)$. The Hardy space H^∞ is identified with l^2. Because L is linear and is an isometric isomorphism between H^∞ and l^2, this implies that H^∞ is a Hilbert space. Using the formal power series inner product, let $f(z) = \sum_{n=0}^{\infty} a_n z^n$ and $g(z) = \sum_{n=0}^{\infty} b_n z^n$ be analytic in the unit disk D. The inner product is $< f, g > = \sum_{n=0}^{\infty} a_n^* b_n$. All sums are from zero to infinity from here on. By the CBS inequality, $|\sum a_k^* b_k|^2$ is less than or equal to $\sum |a_k|^2 \sum |b_k|^2$. So for every z_0 in D, $g_{z_o}(z) = \sum z_0{}^{*n} z^n = 1/(1 - z_0{}^* z)$. Note that $< f, g_{z_o} > = \sum a_k z_0{}^{*n} = f(z_0)$, and $| < f, g_{z_o} > |$ is less than or equal to $\|f\|\ \|g_{z_o}\|$. The results will lead to an evaluation functional. The induced norm is $\|f\|^2 = \sum_{n=0}^{\infty} |a_n|^2$. To see that each power series converges to a function on D, simultaneously, consider the evaluation functional $E_z(f)$. The objective then is to show that it is bounded, for z in D, so let $E_z(f) = \sum_{n=0}^{\infty} a_n z^n$. This is a complex number, so E_z is a functional. The absolute value $|E_z(f)| = |\sum_{n=0}^{\infty} a_n z^n\ |$ is less than or equal to $\sum_{n=0}^{\infty} |a_n \|z|^n$ is less than or equal to $[\sum_{n=0}^{\infty} |a_n|^2 \sum_{n=0}^{\infty} |z|^{2n}]^{1/2}$ that is less than or equal to $\|f\|\ [1/(1 - |z|^2)]^{1/2}$. Accordingly, $\|E_Z\|$ is bounded by $[1/(1 - |z|^2)]^{1/2}$, and H^∞ is a RKHS over X = D. The kernel for a point w in D, in this case, is the Szego kernel: $K(z, w) = K_w(z) = 1/(1 + w^* z)$. To see this, use the definition of the inner product for $g(z) = \sum_{n=0}^{\infty} w^{n*} z^n$ and $f(z) = \sum_{n=0}^{\infty} a_n z^n$, both in H^∞. Here $< g, f > = \sum_{n=0}^{\infty} w^n a_n = f(w)$. This means that g is a reproducing kernel for w in X = D, and therefore, summing the series for $g = \sum_{n=0}^{\infty} w^{n*} z^n$ gives the Szego kernel $K(w, z) = 1/(1 - w^* z)$. Also, $\|E_Z\|^2 = K(z, z) = 1/(1 - z^* z) = 1/(1 - |z|^2)$.#

For every x in X, there is a K_x in H such that for all f in H $< f, K_x > = f(x)$. The function $K(x, y) = < K_x, K_y >$ is called the kernel function corresponding to H. K is positive definite, that is, for all finite collections of vectors v1… vj, in X the gram-type matrix is positive definite. An alternate definition of kernel function x in X, as well as a feature map $\Phi: X \to l^2$, is given by the m tuple, $(\Phi_1 \ldots \Phi_m)$, where $\sum |\Phi_k|^2 < \infty$. In this case, $< \Phi(x), \Phi(y) > = \sum \Phi_k(x)^* \Phi_k(y) = K(x, y)$. From $H^2 \to \Phi_k(z) = z^{k-1}$. Every kernel function can be written as a feature space.

Example 19.3:

For the vector space of polynomials with domain in D, here D is the unit disk in C and the closure in L^2 is the Hilbert space. The associated inner product for z^n and z^m analytic

in D is given by $<z^n, z^m> = \int_D z^{n*} z^m dx\, dy = \int_0^{2\pi} \int_0^1 r^n e^{-in\theta} r^m e^{im\theta} r\, dr\, d\theta$. It equals zero for all m not equal to n, and when they are equal, $<z^n, z^m> = \pi/(n+1)$.

The Bergman kernel is $K(z, u) = 1/[\pi\,(1 - z^* u)^2]$, where $|z| < 1$ and $|u| < 1$. This follows by using power series, and all sums employ index variable going from zero to infinity. To begin, notice that $K(z, u) = \sum [1/\pi\,(n+1)\, z^{n*} u^n]$ should be the reproducing kernel. Proceeding formally, consider $f(u) = \sum am\, u^m$, analytic in D; then $<K(z, u), f(u)> = <K(z, u), \sum an\, u^m> = \sum \sum an\, [1/\pi\,(n+1) <z^{n*} u^n, u^m>] = \sum \sum an\, [1/\pi\,(n+1)\, z^n <u^n, u^m>] = \sum an\, z^n = f(z)$. Since $\sum w^n = 1/(1-w)$, differentiating the last expression with respect to w gives the following: Sum $\sum n\, w^{n-1} = 1/(1-w)^2$ or sum $\sum (n+1)\, w^n = 1/(1-w)^2$. Now substituting in for w gives $\sum [1/\pi\,(n+1)\, z^{n*} u^n] = 1/(1-z^* u)^2$.#

Example 19.4:

The Mittag-Leffler real-valued RKHS of order $q > 0$ consists of functions in $F = \{f(z) = \sum_{n=0}^{\infty} a_n\, z^{qn}$, such that $\sum_{n=0}^{\infty} |a_n|^2 \Gamma(qn+1) < \infty\}$, where the gamma function is $\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$. The kernel is given by the Mittag-Leffler function: $K_q(\lambda, t) = E_q(\lambda^q, t^q)$, where $E_q(z) = \sum_{n=0}^{\infty} z^n/[\Gamma(qn+1)]$. This function is entire and generalizes the exponential function. When $q = 1$, $E_1(z) = e^z$; moreover, when the Bargmann-Fock space is restricted to the positive x-axis, the Mittag-Leffler real-valued RKHS is a consequence. On the other hand, a complexification of the Mittag-Leffler real-valued RKHS will yield a Mittag-Leffler space of entire functions generalizing the Bargmann-Fock space (Rosenfeld et al., 2018).#

Example 19.5:

The Drury-Arveson space is H_n^2, with B the open unit ball in C^n for n larger or equal to two. When $n = 2$, this space is a Hilbert space of analytic functions on B with a reproducing kernel $K_Z(w) = 1/(1 - <z, w>)$. It is a RKHS with inner product $<f, g> = \sum_{a in Z_+^n} (a!/|a|!) b_a c_a,^*$ where $f(z) = \sum_{a in Z_+^n} b_a z^a, g(z) = \sum_{a in Z_+^n} c_a z^a$. This is a generalization of the Hardy RKHS (Jury and Martin, 2018).#

Example 19.6:

The Bargmann-Fock space is a RKHS of entire functions F, in the Hilbert space L^2, with Gaussian measure, namely the inner product in C^n is given by $<f, g> = \pi^{-n} \int f(z)^* g(z) e^{(-|z|^2)} dz$. Specifically, $F = \{f(z) = \sum_{n=0}^{\infty} a_n z^n$, such that $\sum_{n=0}^{\infty} |a_n|^2 n! < \infty\}$. The reproducing kernel is $K(z, w) = e^{wz*}$, and therefore, $\pi^{-n} \int e^{wz^*} g(z) e^{(-|z|^2)} dz = g(w)$ (Hall, 2013; Barbier, 2013).#

In the second reference mentioned above, Jordan algebras are used in exploring Fock space as well as Bargmann transforms. In that reference, the algebra involved real-valued symmetric matrices. It is interesting to mention that in the present text, in Example 3.3), a Jordan-type algebra results. This occurs when the subalgebra of real-valued symmetric matrices is employed in that example. In general, a Jordan algebra is a unital, commutative, and usually nonassociative algebra obeying an additional constraint, besides those MSA identities provided in Section 3.1. This equational constraint involves multiplication using BINE or $\bullet$. It is given by

Jordan identity: BINE (u, BINE (BINE (u, u), v)) = BINE (BINE (u, u), BINE (u, v)).

Again, the Jordan identity: $u \cdot (u^2 \cdot v) = u^2 \cdot (u \cdot v)$. The more succinct notation may be more understandable. The constraint is a sort of associative type law.

Example 19.7:

With the remarks given earlier, consider the carrier set J, of all 2 by 2 real-valued symmetric matrices, u and v. Using BINE (u, v) = (u v + v u)/2, as in Example 3.3), then by substitution, not only is this structure a unital, commutative, nonassociative algebra, but it is also a Jordan algebra.#

Example 19.8: (Barbier, 2013)

Again refer to the original unital, commutative, nonassociative algebra M over C, given in Example 3.3). Of interest is if u is in M, and there is a v in M such that BINE (u, v) = BINE (v, u) = I, then v is said to be an inverse of u. For instance, using u and v provided below, along with u v and v u,

$$\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \quad \begin{vmatrix} 1 & a \\ a & -1 \end{vmatrix} \quad \begin{vmatrix} 1 & a \\ -a & 1 \end{vmatrix} \quad \begin{vmatrix} 1 & -a \\ a & 1 \end{vmatrix}.$$

Then, BINE (u, v) = (u v + v u)/2 = the identity I. For every value of a in C, there exists an inverse for u.#

19.3 Reproducing kernel Hilbert space over R

The function k: X x X→R is a kernel iff there exists a Hilbert space H and a map Φ: X→H such that $<\Phi(x), \Phi(y)>$. Given are X and a kernel function, a scalar product, and a mapping Φ: X→H such that $k(x,y) = <\Phi(x), \Phi(y)>$. To see this, given X and k there exists a mapping, Φ: X→H where $k(x,y) = <\Phi(x), \Phi(y)>$. This exists for all x and y in X. The RKHS is a space of functions. For any x in X, use $k_x = \Phi(x) = k(x, \cdot)$, where we use x as a parameter, and let k_x: X→R, so k_x is a functional. However, every point in X gets mapped into a function. Let S be the range or image of {k_x, such that x is in X}. Take the set of all linear combinations, that is, the span of these functions makes S become a vector space. Define the inner product $<k_x, k_y> = <k(x, \cdot), k(y, \cdot)> = k(x, y)$. This inner product can be extended for arbitrary functions f and g, where $f = \sum_i a_i\, k(x_i, \cdot)$ and $g = \sum_j b_j\, k(y_j, \cdot)$. Here, $<f, g> = \sum_{i,j} a_i b_j k(x_i, y_j,)$, and let S′ be the resulting inner product space. If the completion is performed making every Cauchy sequence converge by adding limit points, a RKHS H is obtained.

The kernel is self-reproducing and positive semidefinite. Use $<K(., x), K(., y)> = K(y, x) = <K(., y), K(., x)>^*$. Thus for the real case, the kernel is symmetric.

Example 19.9:

One of the simplest examples of a RKHS is l^2. Here, these are all functions from the positive integers N into the reals and are square summable. These functions form a vector

space and a Hilbert space with induced norm squared $\|f\|^2 = \sum |f(n)|^2$. An evaluation functional $E_n: l^2 \to R$, for each n in N, is $E_n(f) = f(n) = < f, e_n >$, where e_n in l^2 is such that $e_n(n) = 1$, and for all other values in N, $e_n(m) = 0$. Moreover, $|E_n(f)|$ is less than or equal to $\|f\|$, so it is a bounded functional. As an instance, let f (n) = 1/n; then f is in l^2. As an evaluation functional, notice that $E_3 = 1/3$, and $\|f\| = \pi/6^{1/2}$.#

Example 19.10:

Consider the vector space of affine functions f(t) = a + bt, where a and b are in R. Use the inner product in $L^2(0, 1)$, that is, $<f, g> = \int_0^1 f\, g$. Then to find the kernel, K(t,s), which recreates the affine function, that is, $a + bt = \int_0^1 (a + b\, s)\, K(t,s)\, ds$, first use $1 = \int_0^1 K(t, s)\, ds$, and next use $t = \int_0^1 s\, K(t,s)\, ds$. Substituting into the last two equations $K(t,s) = a(t) + b(t)\, s$ gives $1 = a(t) + b(t)/2$ and $t = a(t)/2 + b(t)/3$. Solving for a(t) and b(t) gives $1 - 2t = -b(t)/6$, so $b(t) = -6 + 12t$ and $a(t) = 4 - 6t$. The kernel is $K(t,s) = 4 - 6(s + t) + 12st$. To verify the result, note that $\int_0^1 (a + b\, s)\, K(t,s)\, ds. = 4a - 3a - 6at + 6at + 2b - 2b - 3bt + 4bt = a + bt$.#

Example 19.11:

Consider band-limited functions f(t) in L^2 that have a Fourier transform $F(w) = \int_{-\infty}^{\infty} f(t) e^{-iwt} dt$, of compact support, on [− a, a], a > 0. Because F(w) is also in L^2 and it is of finite support, then F(w) is also in L^1. Consequently f(t) has numerous properties; see the excellent reference (Goldberg, 1961):

1) Inverse Fourier transform: $f(t) = 1/(2\pi) \int_{-\infty}^{\infty} F(w) e^{iwt} dw = 1/(2\pi) \int_{-a}^{a} F(w) e^{iwt} dw$
2) Uniform continuity: for any $\epsilon > 0$, there is $\delta > 0$ such that $|t - t'| < \delta$ implies $|f(t) - f(t')| < \epsilon$.
3) Bounded: |f(t)| is less than or equal to $1/(2\pi)\|F\|_1$
4) Riemann-Lebesgue Lemma: limit as $|t| \to \infty$ implies $f(t) \to 0$.

The inner product $<f(t), g(t)> = \int_{-\infty}^{\infty} f^*(x)\, g(x)\, dx$. Since f is continuous, a functional $E_x(f) = f(x)$ is defined, and from 3), it is bounded. So H is a RKHS. The reproducing kernel is $K(x,y) = \sin[a(x-y)]/[\pi(x - y)]$.

The Fourier transform of the sinc function g(t) = sin(t)/t, is given by a Cauchy improper integral $\lim_{R \to \infty} \int_{-R}^{R} g(t) e^{-iwt} dt = \pi\, \chi_{[-1,1]}(w)$. By using the scaling and shifting properties of the Fourier transformation, it follows that the Fourier transform of K(x,y) is $e^{-wix}\, \chi_{[-a,a]}(w)$. Finally, by Plancherel's theorem, $<K(x,y), f> = \int_{-\infty}^{\infty} K(x,y)^*\, f(y)\, dy = 1/(2\pi) \int_{-a}^{a} F(w) e^{iwx} dw = f(x)$.#

19.4 Mercer's theorem

For K, in L^2, K: X x X → R is called a positive definite kernel provided that K is symmetric. That is, K(x, y) = K(y, x), and it is positive semidefinite; this means that for any nonzero function f in L^2 it follows that $\int\int f(x)\, K(x,y)\, f(y)\, dx\, dy$ is greater than or equal to

zero. It is positive definite when $\int\int f(x)\, K(x,y)\, f(y)\, dx\, dy > 0$. It will be assumed that K has the finite trace property; here K is continuous and also the integral $\int\int K(x,y)^2\, dx\, dy$ is finite.

Eigenvalues and eigenfunctions arise similar to those in Fredholm-type integral equations. Specifically, ϕ is an eigenfunction for the kernel K if $\int K(x,y)\, \phi(y) dx = \lambda\phi(x)$. Here, λ is an eigenvalue. This equation is denoted as an inner product: $<K, \phi> = \lambda\phi$. Eigenvalues and eigenfunctions provide estimates on errors occurring in learning machines.

A theorem of Mercer states that if K is a positive definite kernel with finite trace, then there exists a countably infinite set of eigenfunctions, $\{\phi_i\}$ with corresponding eigenvalues λ_i where λ_1 is greater than or equal to λ_2 that is greater than or equal to λ_3 ... is greater than zero. Finally, $K(x,y) = \sum_{n=1}^{\infty} \lambda_n \phi_n(x) \phi_n(y)$, and the convergence can be shown to be absolute and uniform. For f in L^2, the eigenfunctions can be employed very similar to Fourier series computations. Here, $\int f(y)\phi_n(y)dx = <f, \phi_n> = f_n$, where ϕ_n is an eigenvector associated with the kernel K, and λ_n is the corresponding eigenvalue. In this case, an inner product can be created in the kernel space involving functions f and g; this inner product is defined as follows: $<f, g> = \sum_{n=1}^{\infty} (f_n \cdot g_n)/\lambda_n$.

Feature maps can be created with the help of Mercer's theorem. Here, the feature space is l^2, and the feature map: $X \rightarrow l^2$, and $\Phi(x) = \lambda^{1/2}(\phi_1(x), \phi_2(x), \ldots)$.

Whenever and if only N of the eigenvalues are positive, then the feature map: $X \rightarrow R^N$. Forming the l^2 inner product will provide the formula for the kernel, $<\Phi(x), \Phi(y)> = \sum_{n=1}^{\infty} (\lambda_n^{1/2}\phi_n(x))(\lambda_n^{1/2}\phi_n(y)) = \sum_{n=1}^{\infty} \lambda_n \phi_n(x) \phi_n(y) = K(x,y)$.

Example 19.12:

This is an instance of a Fredholm-type integral equation of the second kind, since the unknown function appears outside the integral, and within part of the integrand: $\int K(x,y)\, \phi(y)\, dy = \lambda\phi(x)$. Fredholm integrals of the first kind only have the unknown ϕ, within the integrand. In any case, a solution for $\phi(x)$ will be found for $\int_{y=0}^{1} x\, y\, \phi(y)\, dy = \lambda\phi(x) + x$. This integral equation could be solved by differentiating with respect to x. Instead, first multiply the whole equation by x. Then integrate again from zero to one, but this time with respect to x. Then assuming that $\phi(y)$ is smooth enough for Fubini's theorem to hold, the integral with respect to x and with respect to y can be interchanged. Thus, it follows that $\int_{y=0}^{1} y\, \phi(y)\, dy \int_{x=0}^{1} x^2\, dx = \lambda \int_{x=0}^{1} x\, \phi(x)\, dx + \int_{x=0}^{1} x^2\, dx$. This becomes after integrating and re-arranging $[1/3 - \lambda] \int_{y=0}^{1} y\, \phi(y)\, dy = 1/3$, so $\int_{y=0}^{1} y\, \phi(y)\, dy = 1/[1 - 3\lambda]$. Substituting this back into the original equation gives $x/[1 - 3\lambda] = \lambda\phi(x) + x$, and therefore $\phi(x) = x/[\lambda(1 - 3\lambda)] - x/\lambda$. To verify that this is a solution, using it in the original equation gives $\int_{y=0}^{1} x\, y\, \{y/[\lambda(1 - 3\lambda)] - y/\lambda\}\, dy = \lambda\{x/[\lambda(1 - 3\lambda)] - x/\lambda\} + x$. Integrating yields $x/[3\lambda(1 - 3\lambda)] - x/(3\lambda) = x/[(1 - 3\lambda)] - x + x = x/[(1 - 3\lambda)]$. But the left hand side is $x/[3\lambda(1 - 3\lambda)] - x/(3\lambda) = x/(1 - 3\lambda) + x/(3\lambda) - x/(3\lambda)$. Thus showing both sides are equal.#

Example 19.13:

This is probably the most important example of Mercer's theorem. For real-valued functions f(x) periodic, of period 2π, with basic domain $[-\pi, \pi]$, the Fourier series will extend

this function to the whole real line. In this case, $f(x) = \sum_{k=-\infty}^{\infty} f_k e^{ikx}$ where f_k are the Fourier coefficients. Integrating: $\int_{-\pi}^{\pi} f(x)e^{-inx}dx = \int_{-\pi}^{\pi} \sum_{k=-\infty}^{\infty} f_k e^{ikx}e^{-inx}dx$, interchanging the sum and the integral, this gives $\sum_{k=-\infty}^{\infty} f_k \int_{-\pi}^{\pi} e^{ikx}e^{-inx}dx = \sum_{k=-\infty}^{\infty} f_k \int_{-\pi}^{\pi} e^{ix(k-n)}dx = 2\pi f_k$ whenever $n = k$, and zero otherwise. Therefore, $f_k = 1/(2\pi)\int_{-\pi}^{\pi} f(x)e^{-ikx}dx$. The exponentials form an ON basis on $[-\pi, \pi]$. Since f(x) is real, the Fourier coefficients are conjugate symmetric, $f_k* = f_{-k}$. If it is assumed that f(x) is an even function, then $f_k = f_{-k}$. Also assume that the kernel is of the form $K(x, y) = K(x-y)$ and has the aforementioned properties. Let the Fourier representation of the kernel be $K(x) = \sum_{k=-\infty}^{\infty} k_k e^{ikx}$.

Now consider the L^2 inner product of $f(x) = \sum_{k=-\infty}^{\infty} f_k e^{ikx}$ and $g(x) = \sum_{k=-\infty}^{\infty} g_k e^{ikx}$; then $<f, g> = <\sum_{n=-\infty}^{\infty} f_n^* e^{-inx}, \sum_{k=-\infty}^{\infty} g_k e^{ikx}> = \sum_{n=-\infty}^{\infty} f_n^* g_n$. Next a subspace H of L^2 will be a RKHS by defining an inner product $<f, g>_H = \sum_{n=-\infty}^{\infty} (f_n^* g_n)/k_n$. The induced norm in H is $\|f\|_H = <f, f>_H = \sum_{n=-\infty}^{\infty} (f_n^* f_n)/k_n = \sum_{n=-\infty}^{\infty} |f_n|^2/k_n$. So in this RKHS H, it must be that $\sum_{n=-\infty}^{\infty} |f_n|^2$ is convergent. The reproducing property must be shown using $K(x) = \sum_{k=-\infty}^{\infty} k_k e^{ikx}$, so $K(x-y) = \sum_{k=-\infty}^{\infty} k_k e^{ik(x-y)}$. For a function f in H, $<K(,y), f>_H = \sum_{n=-\infty}^{\infty} (k_n e^{-iny})* f_n)/k_n = \sum_{n=-\infty}^{\infty} f_n e^{iny} = f(y).\#$

Example 19.14:

This example will continue from the previous example, Example 19.13. From there, it was seen that the kernel $K(x, y) = K(x-y) = \sum_{k=-\infty}^{\infty} k_k e^{ik(x-y)} = \sum_{k=-\infty}^{\infty} e^{ikx} k_k e^{-iky}$. Similarly $K(x-z) = \sum_{k=-\infty}^{\infty} e^{ikx} k_k e^{-ikz}$. Now it will be seen that the kernel itself can be reproduced: $<K(.,y), K(.,z)>_H = \sum_{n=-\infty}^{\infty} [K(.,y)*, K(.,z)]/k_n = \sum_{k=-\infty}^{\infty} [(k_k e^{-iky})* (k_k e^{-ikz})]/k_n = \sum_{k=-\infty}^{\infty} [(k_k e^{kiy})(k_k e^{-ikz})]/k_n = \sum_{k=-\infty}^{\infty} [(k_k^2 e^{kiy-ikz})]/k_n = \sum_{k=-\infty}^{\infty} (k_k e^{kiy-ikz}) = K(y-z)$. Finally, the feature maps will be identified. Again writing the inner product, $<K(,y), f>_H = \sum_{n=-\infty}^{\infty} ((k_n e^{-iny})* f_n)/k_n$. Use as the feature map $\phi_k(x) = (k_n e^{-inx})$. To see that the feature maps obey the reproducing property, use the inner product $<\phi_k(x), \phi_k(y)> = \sum_{n=-\infty}^{\infty} ((k_n e^{-inx})* k_n e^{-iny})/k_n = \sum_{n=-\infty}^{\infty} k_n (e^{inx} e^{-iny}) = \sum_{n=-\infty}^{\infty} k_n e^{in(x-y)} = K(x, y).\#$

19.5 Spectral theorems

It was seen in Section 13.1 that bounded operators have a nonempty compact spectrum consisting of an eigenvalue spectra, continuous spectra, or residual spectra. In finite dimensions n, every matrix has only an eigenvalue spectrum, and its cardinality is between one and n. This is a result of the fundamental theorem of algebra. In this case, the characteristic equation always has complex roots. Among the main results from Chapter 13 is that for a self-adjoint operator T, in B(H,H), then the eigenvalues are all real. Eigenvectors corresponding to distinct eigenvalues are always orthogonal. Finally, the eigenvalues lie in the interval [m, M] of R. Moreover, from the numerical range development, $m = \inf <v, Tv>$ and $M = \sup <v, Tv>$, both over $\|v\| = 1$, v in H. Finally the residual spectrum, sprt is empty; additionally, the residual spectrum is empty for the larger class of operators, the normal operators.

Again to motivate the spectral theorem involving T, a self-adjoint operator in B(H,H), consider the carrier set R^n. Additionally, let T be a matrix over H with distinct eigenvalues, that is, T has a nondegenerate discrete spectrum. Assume that values within sppT are ordered, that is, $m = \lambda_1 < \lambda_2 < \ldots < \lambda_n = M$. The corresponding eigenvectors should be normalized, thereby forming an ON basis. Define a projection operator P_j, such that $P_j v = < v_j, v > v_j = a_j v_j$, where v_j is the jth eigenvector for T and a_j is in R. Since any vector in H is a superposition of the basis element, then $v = \sum_{j=1}^{n} a_j v_j$, and so $v = \sum_{j=1}^{n} P_j v$. As a consequence, $\sum_{j=1}^{n} P_j$ can be thought to be equal to the identity element I. The sum of projections in this case is often called the resolution of the identity. The operation Tv can be written as $Tv = \sum_{j=1}^{n} \lambda_j P_j v$. Accordingly, when v is nonzero, the operator $T = \sum_{j=1}^{n} \lambda_j P_j$.

At this juncture, it is time to provide the essence of spectral theories from an engineering perspective. Basically in this case, it will be explained what is meant by integrating on the spectrum spT. First, forget the projections for a moment, and consider delta functions, located at each eigenvalue, $\delta(t - \lambda_j)$ for j = 1, 2,. . ., n. For a continuous function f on the real line, a type of spectral integral for f is $\int_{-\infty}^{\infty} f(t) \sum_{j=1}^{n} \delta(t - \lambda_j)\, dt$. Here, the integral goes from minus infinity to plus infinity. It could have instead only went to a finite upper limit, say x in R. In that case, only those eigenvalues less than or equal to x would be included in the integral. This integral operation maps from the reals into the reals; specifically, it maps the spectrum of T into R.

Example 19.15:

In C^2, consider the self-adjoint matrix T =

$$\begin{matrix} |4 & 6| \\ |6 & 9| \end{matrix}$$

The eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = 13$. Consequently, a type of spectral integral occurs, $\int_{-\infty}^{\infty} f(t)\, [\delta(t) + \delta(t-13)]\, dt = f(0) + f(13)$.#

Returning to the actual projection operations, $P_j\, v = < v_j, v > v_j = a_j\, v_j$, where v_j is the jth eigenvector for T, and a_j is in R. The objective in the next example is to actually find these projective operators.

Example 19.16:

Refer to the previous example, and use the operator T along with the associated eigenvalues. The corresponding ON eigenvectors are $v_1 = 1/13^{1/2}\ (-3\ \ 2)'$ and $v_2 = 1/13^{1/2}\ (2\ \ 3)'$. The projection operators in C^2 are such that for any v in C^2, $P_j\, v = < v_j, v > v_j$, for j = 1, 2. So let P represent the generic projection matrix, and let v be the arbitrary vector in C^2. Illustrated below is P, and then following P is the two-by-one column vector v. Finally, the product 2 by 1 column vector P v is given in the third position below:

$$\begin{matrix} |x & y| \end{matrix} \quad \begin{matrix} |a| \\ |b| \end{matrix} \quad \begin{matrix} |xa + yb| \\ |wa + zb| \end{matrix}.$$

To start, P_1 will be determined. But first, only the top row of this matrix, x and y, will be found. Set the first tuple of $P_1 v$, [x a + y b] equal to the inner product of the first

eigenvector, with the vector v, all multiplied by the first tuple of the first eigenvector, $1/13^{1/2}$ [(−3 2) (a b)′] $(-3/13^{1/2})$. Form the inner product above and equate coefficients of a and b, where [x a + y b] = −3/13 [−3 a + 2 b]. Finally, solving for x and y yields x = 9/13 and y = −6/13. This computation provides the first row of P_1.

The second row of P_1 is found in an analogous manner, but using the bottom tuple of Pv, as well as the bottom tuple of the first eigenvector. Therefore, [w a + z b] = $1/13^{1/2}$ [(−3 2) (a b)′] $(2/13^{1/2})$. Now equate coefficients of a and b, where [w a + z b] = 2/13 [−3 a + 2 b], and solving for x and y yields w = −6/13 and y = 4/13. In exactly the same way, the two-step procedure can be performed using the second eigenvector to find the second projection matrix, P_2. Both 13 P_1 and 13 P_2 are illustrated below.

$$\begin{matrix} |9 & -6| & |4 & 6| \\ |-6 & 4| & |6 & 9| \end{matrix}$$

Note that $P_1 + P_2 = I$. Also, $T = \sum_{j=1}^{2} \lambda_j P_j$. This follows by substituting the values of the eigenvalues and the projections; doing this gives $T = 0P_1 + 13P_2$ #

Continuing with the projection operators, these operators are used in mapping the spectrum of T into B(H,H). Consider the staircase-type functions consisting of sums of projections, $E_\lambda = \sum_{\lambda >= \lambda_j} P_j$. Notice that $E_\lambda = 0$ for $\lambda < \lambda_1$, and for λ in the interval $[\lambda_1, \lambda_2)$, $E_\lambda = P_1$; also for λ in the interval $[\lambda_2, \lambda_3)$, $E_\lambda = P_1 + P_2$. And so on, until the values of λ are greater than or equal to λ_n; then $E_\lambda = I$. The operator E_λ can be thought to be a staircase-type function, continuous from the right with saltus s_j, at each eigenvalue. That is, at each eigenvalue, λ_j, when $\lambda = \lambda_j$, the jump value is $P_j - P_{j-1} = s_j$. In general, $s_j = E_{\lambda_j} - E_{\lambda_{j-}}$, where $E_{\lambda_{j-}}$ = limit as $\lambda \to \lambda_j$, from the left.

Now, compare the following conclusion with the Stieltjes integral illustrated in Example 13.29. For a continuous function f on the real line, the spectral integral of f is formally given by $\int_{-\infty}^{\infty} f(t)dE_\lambda = \sum_{j=1}^{n} f(\lambda_j)(P_j - P_{j-1}), P_0 = 0$. In the referenced example, more was going on besides the discrete spectrum. Subsequently, it will be seen that the continuous spectrum can be modeled a lot like the calculations involved in that example. First, an example of an integral over the spectrum will be made resulting in operator-valued results in B(H,H).

Example 19.17:

Refer to Example 19.16, $E_\lambda = \sum_{\lambda >= \lambda_j} P_j$. Then, $E_\lambda = 0$ for $\lambda < \lambda_1 = 0$, and for λ in the interval $[\lambda_1, \lambda_2) = [0, 13)$, $E_\lambda = P_1$, also for λ greater or equal to $\lambda_2 = 13$, $E_\lambda = P_1 + P_2 = I$. The best way to describe E_λ is to partition R into three regions: first, for all $\lambda < 0$, second for λ in [0, 13), and third for λ greater than or equal to 13. Now use the characteristic functions, χ_A = one for λ in A and equal to zero otherwise. Point sets A will be represented as points themselves. Then E_λ is described in each of the regions, as a 2 by 2 matrix consisting of four functions; these function matrices are listed in order:

$$\begin{matrix} |0\chi_{(-\infty,0)} & 0\chi_{(-\infty,0)}| & |9/13\chi_{[0,13)} & -6/13\chi_{(0,13)}| & |1\chi_{(13,\infty)} & 0\chi_{(13,\infty)}| \\ |0\chi_{(-\infty,0)} & 0\chi_{(-\infty,0)}| & |-6/13\chi_{[0,13)} & 4/13\chi_{[0,13)}| & |0\chi_{(13,\infty)} & 1\chi_{(13,\infty)}| \end{matrix}.$$

The saltus at the eigenvalues is given at 0 and 13; these are the jump values. In terms of two-by-two function matrices, these values are given by

$$\begin{matrix} |9/13\chi_0 & -6/13\chi_0| & |4/13\chi_{13} & 6/13\chi_{13}| \\ |-6/13\chi_0 & 4/13\chi_0| & |6/13\chi_{13} & 9/13\chi_{13}| \end{matrix}.$$

So, at the first eigenvalue, $\lambda_1 = 0$, the saltus matrix is $s_1 = P_1 - 0 = P_1$, and the first matrix above is $P_1\ \chi_0$. At $\lambda_2 = 13$, the saltus matrix is $s_2 = P_2 - P_1 = I - P_1$, and the second matrix above is $(I - P_1)\ \chi_{13}$. The saltus matrices and the saltus function matrices are isomorphic; accordingly, in this case, the saltus matrix will only be used herein. Then, $\int_{-\infty}^{\infty} f(\lambda) d\ E_\lambda = f(0)P_1 + f(13)(I - P_1)$. This integral results in a linear combination of operators in B(H,H). It could also be written as $\int_{-\infty}^{\infty} f(\lambda) d\ E_\lambda = f(0)E_{\lambda_1} + f(13)(E_{\lambda_2} - E_{\lambda_1}) = f(0)E_0 + f(13)(E_{13} - E_0) = f(0)s_0 + f(13)s_{13}$. So, for instance, if $f(\lambda) = \lambda e^\lambda$, then $\int_{-\infty}^{\infty} \lambda e^\lambda d\ E_\lambda =$

$$\begin{matrix} |4e^{13} & 6e^{13}| \\ |6e^{13} & 9e^{13}| \end{matrix}.\#$$

Continue for a moment with the assumption that T has a nondegenerate discrete spectrum. The spectral integral of f is formally given by $\int_{-\infty}^{\infty} f(\lambda) d\ E_\lambda = \sum_{j=1}^{n} f(\lambda_j)(P_j - P_{j-1}), P_0 = 0$. It can be seen that this formal integral can be written as a summation of saltus operators, namely $\int_{-\infty}^{\infty} f(\lambda) d\ E_\lambda = \sum_{j=1}^{n} f(\lambda_j) s_j$. Also, an additional spectral representation using $T = \sum_{j=1}^{n} \lambda_j P$ is $T = \sum_{j=1}^{n} \lambda_j s_j$. The integral above is only a symbolic representation for the summation. To obtain a more common meaning of the integral, form the inner product for any v and w in H. Namely, use $< v, T\ w > = \sum_{j=1}^{n} \lambda_j < v, s_j w >$. In this case, the inner product can be given by a true Stieltjes integral: $< v, T\ w > = \int_{-\infty}^{\infty} \lambda\ d\ g\ (\lambda)$, where $g(\lambda) = < v, E_\lambda w >$. Here, the integrator g is a real-valued step-type function, and this is the spectral theorem for the operator T.

Example 19.18:

To obtain a simple example of the actual Stieltjes integral $\int_{-\infty}^{\infty} \lambda\ d\ g\ (\lambda) = < v, T\ w > = \sum_{j=1}^{n} \lambda_j < v, s_j w >$, consider Example 19.17. Here, the eigenvalues 0 and 13 along with the operators T, s_0, and s_{13} are specified. Let $v = |0>$ and $w = |1>$. Then, $< v, T\ w > = (1\ \ 0)\ \ (6\ \ 9)' = 6$. Also, $< v, s_{13}\ w > = (1\ \ 0)\ \ (6/13\ \ 9/13)' = 6/13$, and therefore, $\sum_{j=1}^{2} \lambda_j < v, s_j w > = 0 + 13 \cdot 6/13 = 6.\#$

The spectral theorem for self-adjoint bounded operators on a Hilbert space is almost exactly the same as the one mentioned right before the last example. The major difference is that in place of the saltus, a measure is employed. The measure is described in detail in Appendix A.2. As in the previous cases, the integrals are the Stieltjes integrals over the spectrum. The need for a spectral measure is that the continuous spectrum, as well as limit points arriving from clusters of discrete spectral points, can also be included in the spectral integration. This is somewhat like the integration process illustrated in Example 13.29.

For T self-adjoint and bounded with spectrum spT, there corresponds a spectral measure u. It is such that, for all v and w in the Hilbert space H, the inner product $< v, Tw > = \int_{spT} \lambda d < v, u_\lambda w >$.

Generalization of the aforementioned concepts leads to the projective-valued measure (PVM). These measures have values that are self-adjoint projections. As usual, measures

are defined on a measurable space. The spectral theorem involving PVM associates a PVM with any self-adjoint operator T in B(H,H). Here, $T = \int \lambda dE_\lambda$. The integral is understood as a Stieltjes or Lebesgue integral on the real line. A rigorous well-explained description of PVM, along with direct integrals, can be found in Hall (2013).

Example 19.19:

Let T be a bounded self-adjoint operator on the Hilbert space with carrier set $H = R^2$. In addition, assume that $|v>$ is a unit vector in H and that f is in BM, that is, f is bounded and measurable. The inner product $<v|f(T)\ v>$ is a positive functional on the spectrum $spT = \sigma(T)$. As such, there exists a unique spectral measure u_v associated with $|v>$. The following equality holds on the compact set $\sigma(T)$, where $<v|f(T)v> = \int_{\sigma(T)} f(\lambda) du|v>_v$, and the integral is of Lebesgue type. Use for $|v> = 2^{-1/2}(|0> + |1>)$, which is also in R^2 being the column vector, $2^{-1/2}\ (1\ 1)'$. Assume that $f(t) = t^2$ and that T is given by the following matrix:

$$\begin{matrix} |4 & 0| \\ |0 & 2| \end{matrix}$$

Accordingly, $spT = sppT = \sigma(T) = \{4, 2\}$. In this case, $<v|f(T)\ v> = <2^{-1/2}\ (1\ 1)', T^2\ 2^{-1/2}\ (1\ 1)'> = 10$. The PVM is $\int_{\sigma(T)} f(\lambda) du_v|v> = [f(4)u_v(\{4\}) + f(2)u_v(\{2\})]|v>$. Here, $f(4) = 16$ and $f(2) = 4$; the projections $u_v(\{4\})$ and $u_v(\{2\})$ are given as follows:

$$\begin{matrix} |1 & 0| \\ |0 & 0| \end{matrix} \quad \begin{matrix} |0 & 0| \\ |0 & 1| \end{matrix}.$$

Substitute these values into $\int_{\sigma(T)} f(\lambda) du_v|v> = 16u_v(\{4\})2^{-1/2}(11)' + 4u_v(\{2\})2^{-1/2}(11)' = 2^{-1/2}(164)'$. Note that the Borel sets in this example are in $\{\phi, \{4\}, \{2\}, \{4, 2\}\} = 2^{\sigma(T)}$.#

This section will end with a most important spectral theorem that is easy to understand. It is for self-adjoint compact operators T, in a Hilbert space H. Using N, the null space of T, then the dimension of $N^\perp$ is at most countable and has an ON basis $\phi_n, n = 1, 2, \ldots$, of eigenvectors of T. Here, $T\phi_n = {}_n\phi_n$, where for every n, there is a nonzero eigenvalue. Moreover, if $N^\perp$ is not finite, then the eigenvalues λ_n go to zero, that is, $\lambda_n, \to 0$ as $n \to \infty$, and this is the only accumulation point. Additionally, the eigenspaces for nonzero eigenvalues are finite dimensional.

Since nonzero eigenvalues are isolated points in the complex plane, Cauchy integral theorem provides a Riesz projection $E(\lambda)$. This is done by integrating over a SCROC, enclosing only a single eigenvalue. λ A SCROC is a simple closed rectifiable oriented curve. These operators are spectral projections and also $E(\lambda) \cdot E(\mu) = E(\lambda)$ whenever $\lambda = \mu$, and zero otherwise (Conway, 1990).

19.6 The Riesz-Markov theorem

Let X be a locally compact Hausdorff space, with a state S in $C^0(X, R)$. Then there exists a unique Borel probability measure u_S such that for all f in $C^0(X, R)$ it follows that $S(f) = \int_X f\ d\ u_S$. Moreover, the functional S(f) can be viewed as being the expected value of

an observable f in state S. Thus, S(f) can be considered the average value. Likewise the conditional variance $\sigma_S{}^2 = S((f - S(f))^2) = S(f^2) - S(f)^2$.

Example 19.20:

Heisenberg uncertainty principle: For the fixed state S, and observables p and q, $\sigma_S(p)\sigma_S(q)$ is greater than or equal to h/2. Here p and q denote position and momentum. Without loss of generality, assume that the means of p and q are zero. So, $\sigma_S{}^2(p)\sigma_S{}^2(q) = S(p^2)S(q^2)$. Since all observables are self-adjoint, it follows that $(ap + ibq)^* = ap - ibq$ for all a, b in R. Since the states are positive, the expression $S[(ap - ibq)\ (ap + ibq)] = a^2\ S(p^2) - ibaS(qp) + iabS(pq) + b^2S(q^2)$ is greater than or equal to zero. The terms involving i can be written using the commutator, that is, it could be written as iabS([p, q]). Let v be the column vector (a b)′, and M =

$$\begin{vmatrix} S(p^2) & 1/2S(i[p,q]) \\ 1/2S(i[p,q]) & S(q^2) \end{vmatrix}.$$

It follows that the aforementioned inequality can be written as v′ M v, which is greater than or equal to zero, and so M is positive definite. Accordingly, the determinant of M is greater than or equal to zero, that is, $S(p^2)S(q^2) - 1/4(S(i[p,q]))^2$ is greater than or equal to zero. Consequently, $S(p^2)S(q^2)$ is larger or equal to $1/4(S(i[p,q]))^2$. Finally, see Section 8.7, where [p, q] = ih, and so $\sigma_S(p)\sigma_S(q)$ is greater than or equal to h/2. This is the Heisenberg uncertainty principle.#

19.7 Some nonseparable Hilbert spaces

Throughout this document, all Hilbert spaces were assumed to be separable. That is, the Hilbert space possesses a countable dense subset, and accordingly, it has a Schauder basis. Nonseparable Hilbert spaces are employed in condensed matter physics. Also in quantum field theory, dropping Poincare invariance for curvature results in a nonseparable Hilbert space. Additionally, in Section 14.10, it was shown that the Weyl version of the CCR C* algebra is not separable. This C* algebra is faithfully represented on the symmetric Fock space.

A couple of other examples of nonseparable Hilbert spaces are given below. In the next section, one of the most important facts about separable Hilbert spaces is proved.

Example 19.21:

Consider a modification of $l^2(R)$, the Hilbert space of all converging square summable sequences of real numbers. Let f: R→R, where f(x) = 0 for all x except for a countable set, and here the sum over all x of $f(x)^2$ is finite. Then the characteristic function of a single point p or q in R, p not equal to q, will be designated by χ_p and χ_q, respectively. Let f (x) = g(x) = 0, except that $f(x) = \chi_p$ and $g(x) = \chi_q$. These functions are in this Hilbert space, and their individual sum squared is one. The distance, however, between them is

$$d(f(x), g(x)) = \left[\text{the sum}, \sum\nolimits_x (f(x) - g(x))^2\right]^{1/2} = \left[\left(f(p) - g(p)\right)^2 + \left(f(q) - g(q)\right)^2\right]^{1/2} = \left[\chi_p^2 + \chi_q^2\right]^{1/2} = 2^{1/2}.$$

There cannot be a countable dense subset of this space. For using any open set at f of radius less than $2^{1/2}$, it only contains f itself, and there are uncountable such points.#

Example 19.22:

The space of all Bohr's almost periodic functions forms a nonseparable Hilbert space. The inner product is $<f, g>$ equals the limit $T \to \infty$, $1/(2T) \int_{-T}^{T} (f(t)^* g(t))\, dt$. The uncountable set $\{e^{iwt}\}$ for w nonzero in R is an orthonormal family of vectors in this space. They are periodic, and moreover, limit $T \to \infty, 1/(2T) \int_{-T}^{T} e^{-iat} e^{ibt} dt = 1$ whenever a = b and equal to zero otherwise. The completion is a nonseparable Hilbert space.#

19.8 Separable Hilbert spaces are isometrically isomorphic to l^2

The proof of the fact that every separable Hilbert space H is isometrically isomorphic to l^2 is given in the following eight steps and is shown for when SCALAR are the reals.

1) Since H is separable, it has a Schauder basis: $\{e_n\}$, n = 1, 2, ….

2) For v in H and T such that $T:H \to l^2$, let T(v) be the sequence consisting of the inner product of v with the ON basis $\{e_n\}$. Denote the sequence by (e_n). Thus, $T(v) = < v, e_1 >, < v, e_2 >, \ldots = (< v, e_n >)$.

3) Utilizing Bessel's inequality given next shows that $T(v) = (< v, e_n >)$ is in l^2.

The Bessel's inequality holds for arbitrary inner product space. It involves the ON basis $\{e_n\}$, along with v. To begin, the sums in the following go from n = 1 to N, $\|v - \text{sum}, \sum(< v, e_n > e_n)\|^2 = < v - \text{sum}, \sum(< v, e_n > e_n), (v - \text{sum}, \sum(< v, e_n > e_n) > = \|v\|^2 - \text{sum}, \sum |< v, e_n >|^2$ is greater than or equal to zero, and so $\|v\|^2$ is greater than or equal to sum, $\sum |< v, e_n >|^2$. Next let $N \to \infty$. This shows that $\|T(v)\|_2 < \infty$, so $T(v) = (< v, e_n >)$ is in l^2. All limits and indices are from n = 1 to ∞, from here on in.

4) Next to show that T is linear, for v and w in H and a in R, note that $T(v + a w) = (< v + a w, e_n >) = (< v, e_n > + a < v, e_n >) = (< v, e_n >) + a (< v, e_n >) = T(v) + a T(w)$

5) Using the ON basis $\{e_n\}$, the unique representation $v = \text{sum}, \sum (< v, e_n > e_n)$ holds. With sums from 1 to ∞, $\|v\|^2 = < \text{sum}, \sum (< v, e_n > e_n), \text{sum}, \sum (< v, e_k > e_k) > = \text{sum}, \sum (< v, e_k >^2 < e_k, e_k >) = \text{sum}, \sum (< v, e_k >^2) = \|T(v)\|^2$. This shows that T is bounded, and moreover, it is an isometry.

6) T is 1–1, since if T(v) = T(w), then $(< v, e_n >) - (< w, e_n >) = 0$, and $(< v - w, e_n >) = 0$, and because for all n, $< v - w, e_n > = 0$; this implies that v = w.

7) T is onto because if (z_n) is any element in l^2, then consider $z = \text{sum}, \sum (z_n e_n)$; it will be seen that z is in H and $T(z) = (z_n)$. The two norm squared, $\|z\|^2 = < z, z > = \text{sum}, \sum (|z_n|^2) < \infty$. So, $T(z) = (< z, e_k >) = (< \text{sum}, \sum (z_n e_n), e_k >) = (z_k \|e_k\|) = (z_k)$.

8) Finally, using the bounded inverse theorem, see Appendix A.6. That is, if A: X → Y, with X and Y Banach spaces and A, 1–1 and onto, then A^{-1}: Y → X is also bounded. Since all these conditions apply, T has a bounded inverse. As such, T is an isometric isomorphism between H and l^2.

References

Aronszajn, N., 1950. Theory of reproducing kennels. Trans. Amer. Math. Soc.
Barbier, S., 2013. The Quantum Mechanical Segal-Bargmann Transform Using Jordan Algebras. Universiteit Gent.
Conway, J., 1990. GTM A Course in Functional Analysis. Springer.
Goldberg, R., 1961. Fourier Transforms. Cambridge University Press.
Hall, B., 2013. Quantum Theory for Mathematicians. Springer.
Jury, M., Martin, R., 2018. Extremal Multipliers of the Drury-Arveson Space. University of Florida.
Katznelson, Y., 1976. 978-0-486-63331-2 Introduction to Harmonic analysis. Dover Pub.
Krantz, S., 2007. Boundary behavior of holomorphic functions: global and local results. Asian J. Math. 11 (2).
Putnam, I., Lecture Notes On C* algebras, 2019.
Rosenfeld, J., Russo, B., Dixon, W., The Mittag Leffler reproducing kernel Hilbert spaces of entire and analytic functions. J. Math., Elsevier, 2018; 463: 576-592.

APPENDIX

A

Hilbert space of wraparound digital signals

The set of all wraparound digital bound vectors and bound matrices both form Hilbert spaces. The inner product is evaluated on the intersection of their co-zero sets. For the empty intersection, this product is zero. Completeness follows using Cauchy sequences (CSs). In these cases, it was shown that they form CSs of bound vectors or bound matrices f_n. The objective is to show f_n converges, say to f, in the one norm; other norms follow in the same manner. The first thing to notice is that if $f_n \to f$, each element v_n within f_n will converge to say v. The elements v_n are tuples within the bound vector, or matrix entries v_n, within the bound matrix f_n. Convergence in this manner follows since $|| f_n - f ||$ is greater than or equal to the difference within specific entry values $| v_n - v |$. Conversely, if every tuple element converges, this means that f_n converges, because there are only a fixed finite number of tuples making up the bound vector or bound matrix structure. As mentioned earlier, the one norm will be employed in the following, but the results hold for any norm. The sequence of facts is proven (Apostle, 1974; Rudin, 1987):

1. Every sequence of real numbers has a monotonic subsequence.
2. Every CS is bounded.
3. A bounded monotonic sequence of reals always converges.
4. Every CS of reals converges.
5. The space of wraparound bound vectors and matrices is complete and therefore forms a Banach and also a Hilbert space.
6. Every sequence v_n of real numbers has a monotonic subsequence. This results follows by using peaks. A peak v_m is an element of the sequence such that v_m is larger than all elements v_n, in the sequence with $n > m$. If there exists an infinite set of peaks, then this is a subsequence consisting of peaks and is strictly decreasing. On the other hand, say that there only exists a finite number of peaks. Let v_m be the last peak. Then start from the $m + 1$st value v_{m+1} in the sequence. This value must be less than or equal to all the following values of this sequence except a finite number for otherwise, this too would

be a peak. Using induction, let v_k be a member of the sequence with $k > m+1$, and v_k is greater or equal to v_{m+1}. Now v_k is not a peak, so there is a value $p > k$ with v_p greater than or equal to v_k. Continuing thusly, the sequence: $v_{m+1}\ v_k\ v_p \ldots$ forms a subsequence of v_n, which is monotonically increasing.

Example A.1:

The sequence: 1 0 1/2 0 1/3 0 1/4 …. This sequence has peaks at every odd position and therefore results in a monotonically decreasing subsequence, 1 1/2 1/3 1/4 ….#

Example A.2:

Consider the sequence of real numbers: 3, 1, 2, −1*, −1/2, −1/3, −1, −1, −1/2**, −1/3, −1/4,…, −1/*n*,… Then peaks exist at the values 3 and 2, and that's it; from this point onward, there are no more peaks. Starting at v_4, the value −1* is less than or equal to all the following terms in the sequence. The element $v_9 = -1/2^{**}$ is greater than $v_4 = -1^*$. Also the value $v_9 = -1/2^{**}$ is not a peak, so an index value larger than 9 will be considered as being part of a monotonic increasing sequence. A subsequence of v_n that is monotonically increasing is thus found. Here, this subsequence is $v_{m+1}\ v_k\ v_p \ldots, = v_4\ v_9\ v_{10} \ldots = -1^*\ -1/2^{**}\ -1/3 \ldots -1/n \ldots$.

2. Next, assuming that v_n is a CS it is easy to see that it is bounded. This follows, because for any positive integer n, $k > N-1$, where N is fixed, implies $|\ v_n - v_k\ |$ is less than or equal to ϵ. In particular, then by the triangle inequality, $|\ v_n\ | - |\ v_N\ |$ is less than or equal to $|\ v_n - v_N\ |$, because $|\ v_n\ | = |\ (v_n - v_N) + v_N\ |$ is less than or equal to $|\ (v_n - v_N)\ | + |\ v_N\ |$. Consequently, $|\ v_n\ |$ is less than or equal to $|\ v_N\ | + \epsilon$, for all $n > N-1$, and for $n < N$, there are a finite number of values of n. An overall bound will be to employ the maximum of a finite number of values, that is, use $\max(|\ v_N\ | + \epsilon, |\ v_1\ |, |\ v_2\ |, \ldots, |\ v_{N-1}\ |)$.
3. In the following, it will be shown that a bounded monotonic sequence of reals v_n always converges. Assume that the sequence v_n is monotonic increasing. Then, let s = supremum of v_n; then for every $\epsilon > 0$, there exists $N > 0$, such that $v_N > s - \epsilon$. This is true since, otherwise, $s - \epsilon$ would be a supremum of v_n. Using monotonicity implies that v_n is greater than or equal to v_N, and for $n > N$, $s - \epsilon < v_n < s + \epsilon$, which is the definition of a sequence converging, thereby showing that v_n converges to s.
4. Next, show that every CS of reals converges. Notice that $|\ v_n - v_{ki}\ | < \epsilon$ holds true for v_{ki}, where v_{ki} is a monotonic subsequence of v_n, because the sequence is CSs. Using the triangle inequality gives $|\ v_n - s\ |$, which is less than or equal to $|\ v_n - v_{ki}\ | + |\ v_{ki} - s\ |$. For N large enough, and n, $ki > N$, since the first expression is less than ϵ since it is CS and the second absolute value term is also less than ϵ, using (3) above gives $|\ v_n - s\ | < 2\epsilon$.
5. Finally, for bound vectors in Wn, say $f_j = \left(a_j b_j \ldots d_j\right)_k^{Wn}$. Let f_j form a CS of bound vectors. So for N large enough and j, $m > N$, it follows that $||\ f_j - f_m\ || < \epsilon$.

Using the one norm $|| f_j - f_m ||_1 = | a_j - a_m | + | b_j - b_m | + \ldots + | d_j - d_m |$ shows that each tuple itself is a CS. Since these are real sequences, they all converge. Say that $a_j \rightarrow a$, $b_j \rightarrow b$, $\ldots$, $d_j \rightarrow d$. Then, $f_j \rightarrow f = (a\, b \ldots d)_k^{Wn}$. The same exact proof holds for bound wraparound matrices. These structures are therefore Banach spaces.

The two norm also provides the same conclusion. Accordingly, these structures form Hilbert spaces. Moreover, the proof above holds for complex-valued wraparound bound vectors and bound matrices. The verification is to just use the real and imaginary parts of the complex quantities.

References

Apostle, T., 1974. Mathematical Analysis, 2ed. Pearson Pub Co.
Rudin, W., 1987. 0070542341 Principles of Mathematical Analysis. Mc Graw Hill.

This page intentionally left blank

APPENDIX

B

Many-sorted algebra for the description of a measurable and measure spaces

An axiomatic framework for measure theory in the many-sorted algebra (MSA) is essentially the same model as is often used in real analysis and integration theory. However, a slight modification is made in the case of an MSA description. For the concept of a measurable space, begin with X a nonempty set. Let S be the set of all subsets of X. That is, $S = 2^X$. Let M be the class of sets that form a sigma algebra of subsets of X. These are specified and are special subsets, which are elements in S. The measurable space is often written as (X, M). The polyadic graph in Fig. B.1A illustrates the two sorts involved in a measurable space description. These sorts are given next along with descriptions and abbreviations. The sorts are the following:

SUBSET-S, which are all subsets of X, thus $S = 2^X$.

M-SET-E, which are all sets in a sigma algebra described later and are special elements from S. M-SET is an abbreviation for measurable sets.

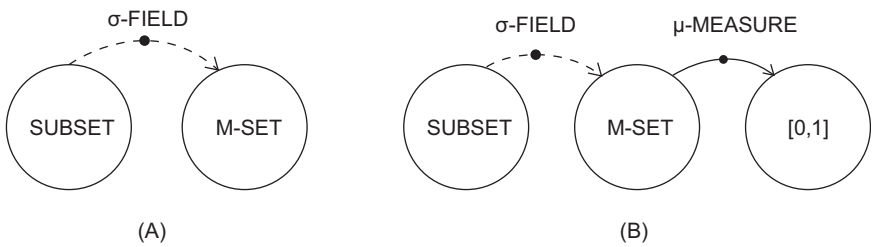


FIGURE B.1 (A) Measurable space and (B) measure space.

There exists a single unary function in a measurable space, and it is called a sigma algebra or a sigma field, σField. It is such that

$$\sigma\text{Field:SUBSET} \rightarrow \text{M-SET}.$$

This function and associated sorts are more easily described by σF: S $\rightarrow$ M. It is a partial (identity) function, abbreviated by σF. It selects which elements or subsets in M are to be called measurable. This is similar to the inverse function for a field structure. The latter partial operator selects nonzero elements to have an inverse. Note that in the polyadic graph, the arrow for σF is dotted, indicative of a partial operation.

The equational identities must hold for all elements E in M:

1. The empty set E = ϕ in S must be in M, that is, $\sigma F(\phi) = \phi$.
2. For E such that σF(E) = E, the complement of E in X, that is, X − E, σF(X − E), is also in M, so σF(X − E) = (X − E).
3. If E_k in S, and $\sigma F(E_k)$ is in M where k = 1, 2, 3, . . ., that is, $\sigma F(E_k) = E_k$, then σF(union of all E_k) = union of all E_k.

Note that, by using (1) and (2), X is also a measurable set, that is, σF(X) = X.

By De Morgan's law, for $\sigma F(E_k)$ in M, σF(intersection of all E_k) is also in M. Point (3) along with this result says that (X, M) is closed under countable unions and countable intersections. The next example illustrates a fundamental measurable space underlying all of Lebesgue integration theory (Halmos, 1950).

Example B.1:

Let X be a locally compact Hausdorff space. A Borel set E is a subset of X such that E is the image of the smallest σF using closed sets in X. These are called the Borel sets belonging to the smallest σ algebra generated by the closed sets of X. To make the example more concrete, let X be the real line R. A Borel set E is a subset of X, or an element of $S = 2^X$, having the property that σF(E) is in M. The empty set and the whole real line R belong in M. The interval closed interval [0, 1] is in M, since [0, 1] is a closed set in R. For the same reason, the interval, [1.5, 2] is in M. Moreover, using constraint number (3), the union of these two sets is in M. Remember that σF is like a partial identity function; it chooses which types of sets within the power set 2^X of X are to be measurable sets. Then, using the union, difference, and intersection closure conditions specified by the constraint conditions, all the other measurable sets can be found. However, constraint number (3) must hold not only for the finite case but also for the countably infinite case. Thus it follows that [0, 1) is also a Borel measurable set. Note, using Borel measurable sets $E_k = [0, 1-1/k]$, k = 1, 2, . . . and then taking the infinite union of E_k gives the desired result. Accordingly, all open sets, half open and half closed, are also in M and are therefore Borel measurable.#

In R^n or C^n, a similar conclusion holds using closed sets in the real case and using these sets for the real and imaginary parts in the complex situation.

Example B.2:

Again, let X be a locally compact Hausdorff space. A Baire measurable set X is a subset of X belonging to the smallest σ algebra generated by specific compact sets K of Ω. These compact sets K are specified to be the intersection of a countable number of open sets in X. Again, in order to provide a more concrete example, let X = R. Then, a typical compact set used in generating a Baire measurable set in this σ algebra is the closed and bounded set K = the intersection of all $A_k = (0-1/k, 1 + 1/k)$, k = 1, 2, ... The intersection results in the compact set K = [0, 1]. When the topology of X is first countable, that is, when there exists a local countable base, the Baire and Borel sigma algebras are equal. Except for rare exceptions, a countable basis always exists, and so in most practical situations, Baire and Borel sigma algebras are the same (Arveson, 1996).#

An instance of a nonfirst countable space is provided in Figure 10.4. In this diagram, there exist an infinite number of loops at the origin. Intuitively, this makes it impossible to find an open ball about the origin fitting into this region.

For a measure space, there exist three sorts: SUBSET, M-SET, and $R+^{\infty}$; here, the first two sorts are as is in a measurable space. The third sort is $R + ^{\infty} = [0, \infty]$. The MSA description of a measure space involves two signature sets ignoring the strict total ordering of signature sets associated with the real numbers in the extended nonnegative real line. The first signature set was specified above containing σField. The second signature set has the unary operator name μMeasure. It is such that

μMeasure is a mapping from M-SET $\rightarrow$ $R + ^{\infty}$. This is illustrated in Fig. B.1B.

Next, use μ in place of μMeasure; the equational relations and identities must hold for the measure μ.

1. μ (X) is in $[0, \infty]$
2. For E_k in M, where k = 1, 2, 3, ... and they are mutually disjoint, that is, their intersection is empty, $\mu(\cup E_k) = \sum \mu(E_k)$.

A measure μ is called sigma finite, which means that for any E in M, E equals the countable union of E_k, for E_k in M, and such that $\mu(E_k) < \infty$, for all, k = 1, 2, 3, ... A measure μ is said to be finite whenever $\mu(X) < \infty$. It is said to be semifinite if for each E in M with $\mu(E) = \infty$, there exists a measurable set, F a subset of E such that $0 < \mu(F) < \infty$. A sigma finite measure is always a semifinite measure. This assertion follows using proof by contradiction. Assume that μ is not semifinite. So there exists a set E in M with $\mu(E) = \infty$, and there is no subset F of E where $0 < \mu(F) < \infty$. Then for μ, being sigma finite, there is a sequence of measurable sets E_k, where X equals the countable union of E_k. It follows that E equals the countable union of [E intersect E_k]. This implies that μ (E) is less than or equal to $\sum_{k=1}^{\infty} \mu$(E intersect E_k). Furthermore, μ(E intersect E_k) is less than or equal to $\mu(E_k)$, which is finite. But since, by assumption, any subset of E has measure zero, and (E intersect E_k) is a subset of E; then μ(E intersect E_k) must equal zero. It follows that $\mu(E) = 0$, giving a contradiction.

Example B.3:

Consider the measurable space (X, M). With X nonempty, $M = 2^X$, and f: X → [0, ∞]. The function f determines a measure μ on M, where $\mu(E) = \sum_{xinE} f(x)$. This is a semifinite measure when and only when $f(x) < \infty$, for all x in X. It follows that if $f(x) = \infty$, then {x} is in M, and there is no proper subset of a single point. In this case, one cannot find a subset of finite measure. Assume that f (x) is finite for all x in X. If $\mu(E) = \infty$, then for a single point x in E, it follows that {x} is a subset of E for which $\mu(\{x\}) = \sum_x f(x) < \infty$.

Example B.4:

The Lebesgue measure on the real line is sigma finite, but not finite. Use the Borel measurable sets $[n, n + 1)$ for all integers n. This provides a disjoint union of countably many unit Borel intervals covering the whole real line. Thus the Lebesgue measure is sigma finite.#

References

Arveson, W., 1996. Notes on Measure and integration in Locally Compact Spaces. U of C, Berkely.
Halmos, P., 1950. Measure Theory. Springer.

APPENDIX

C

Elliptic curves and Abelian group structure

Working with elliptic curves (ECs), the objective is to find all integer or rational solutions for that cubic equation in R^2. Consider the following: $b^2c = 4a^3 - 4ac^2 + c^3$, with c nonzero. Let $x = a/c$ and let $y = b/c$, then the equation becomes the elliptic equation: $y^2 = 4x^3 - 4x + 1$ (see Fig. C.1). Factoring gives the equation $(y - 1)(y + 1) = 4x(x + 1)(x - 1)$. Among the solutions, that is, points on the curve, for this equation, are (0,1), (0, − 1), (1,1), (1, − 1), (−1,1), (−1, − 1). Other solutions, if any, can be found by creating a straight line through two distinct solution points and then solving for the third point of intersection. Since the equation is symmetric about the x-axis, it follows that for any solution the opposite branch will also be a solution for the same abscissa value. The process of choosing two points to determine the third point produces a binary operation; for convenience, it will be called addition.

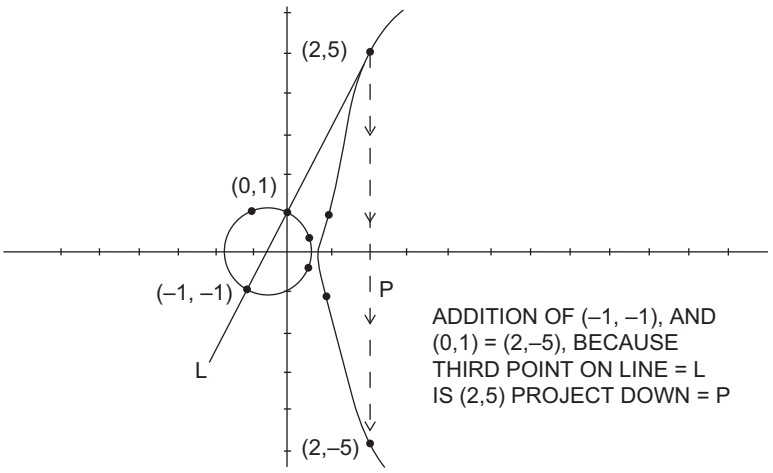


FIGURE C.1 Elliptic curve addition.

Referring to the figure, the line $y = 2x + 1$ goes through (0,1) and $(-1, -1)$ and intersects the elliptic curve at (2,5). Accordingly, (2,-5) is also a point on the curve. The reflected point is the solution and is called the addition of (0,1) and $(-1, -1)$. So the sum of (0,1) and $(-1, -1)$ is $(2, -5)$ (see Fig. C.1). This time, passing a line through (0,1) and $(2, -5)$ gives $y = -3x + 1$, finding the intersecting point yields (1/4,1/4). Consequently, the point $(1/4. -1/4)$ is the solution. Thus (0,1) plus $(2, -5) = (1/4. -1/4)$. Continuing, the line $y = x$ goes through $(-1, -1)$ and (1,1) and yields (1/4.1/4) as before.

The mechanics behind the Abelian group structure for elliptic curves is as follows: The sum of two points on the curve is found by utilizing the equation of a straight line passing through these two points, then finding where it again intersects the curve, thus determining a third point on the given line, and finally, reflecting this found point about the x-axis. If the line through two points is vertical, the third point is ∞, and the point at infinity is therefore the sum. Thus, by including the point at infinity, the structure is a pointed set. Tangents to the curve are handled similarly; in this case, the point of tangency is added to itself. If the tangent line intersects the curve, for a third time, then the sum is that point with the negative ordinate value applied. When there is no intersection with the curve, the point at infinity ∞ is used. The set of points under this binary operation so far is a groupoid (see Fig. C.2).

To see that it is a semigroup, associativity must prevail. Here, returning to projective lines and projective space, a theorem (Cassels, 1991) is applied: Whenever a cubic curve goes through eight of nine points of intersection with two other cubic curves, then a ninth point of intersection must be included. This is often called the ninth point lemma or the Cayley–Bacharach theorem. The proof of associativity follows from this result (Cayley, 1889) (refer to Fig. C.3).

The basic idea is finding two curves C1 and C2 which share eight distinct points with a given curve C, then showing that these curves are cubic and not equal to C, and then applying the aforementioned theorem illustrating that the ninth point is in both C1 and

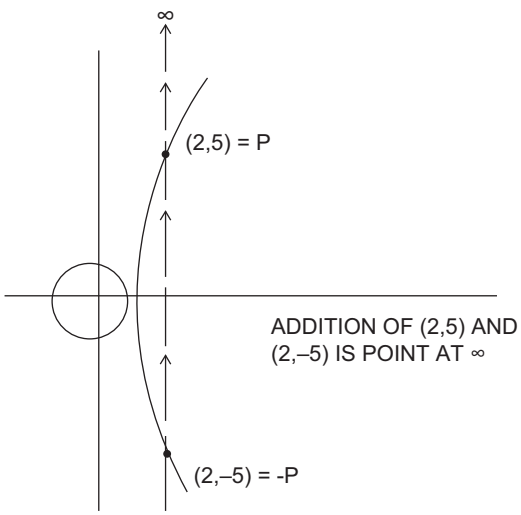


FIGURE C.2 Addition of vertical points on an elliptic curve.

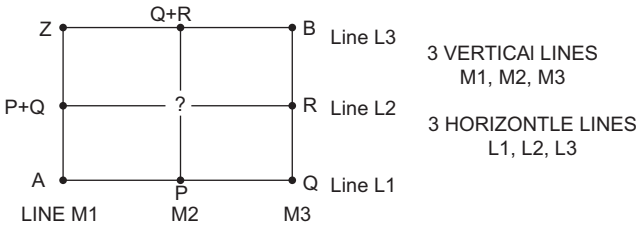


FIGURE C.3 Proof of associative law.

C2. By judicially configuring the eight points in a strategic pattern, containment of the ninth point will prove the associative law. In this figure, the symbol (?) is supposed to be the ninth point. The eight points which are shared by all six straight lines are: A, P, Q, R, B, Q + R, Z, and P + Q. Points on a straight line depict lines in affine space. The overall objective is to show that (P + Q) + R = P + (Q + R); this will follow because line L2 and line M2 intersect at ?. The three lines L1, L2, and L3 each can be given as degree-one homogeneous polynomials: |1 = |2 = |3 = 0. In this case, C1 = |1|2|3 describes a degenerate cubic. Analogously let C2 = m1m2m3, where m1 = m2 = m3 = 0 are degree-one homogeneous polynomials describing M1, M2, and M3, respectively. So given the elliptic curve C, including the eight points, applying the Cayley–Bacharach theorem shows the ninth point (?) is included and in the intersection of L2 and M2. This shows that the structure is a semigroup. Next, showing that the structure is a monoid follows.

A ZERO must be found and prove that it satisfies the zero law to obtain a monoid. The zero is the point at infinity ∞. It obeys the zero law because if you take any single point p, then the sum of p + Z is found by taking the vertical line and connecting these two points. Then for a nontangent point p, there exists a third point q on this same line, thus the sum is − q. However, − q is p. When p is a tangent point and the tangent is vertical, then p + Z = p. The next to show is that this structure is a group. For a group, a MINUS is needed, and the minus identity must be satisfied. However, for point p = (a, b), MINUS (p) = (a, − b) denoted − p for short. Moreover, when these two points are connected with a straight line, the line is always vertical, so p + (−p) = ∞ as desired. Since there is only one point at infinity, MINUS(Z) = ∞. Finally, the group is Abelian because the order of choosing two points, p and q, on a line does not change the resulting third point r, on the same line, which is reflected and consequently p + q = −r = q + p.

Lastly, a theorem by Mordell specifies the group structure for rational solutions on an elliptic curve with rational coefficients. It states that it is a finitely generated Abelian group provided that it is nonsingular, that is, it has distinct roots (Silverman, 1986). So by this theorem, there exists a finite set of points Pj, in this group such that every point P in the group can be expressed as P = summation j = 1 to n of nj Pj for nj in Z. Moreover, the structure of this group was proved by Mazur (Dumber, 2019) to be of the form T x Z x Z x ... x Z, where T is a torsion subgroup with order at most 16, and T always being one of fifteen possibilities. Additionally, there are r copies of Z above, and it is called the rank. Finally, for an elliptic curve $y^2 = x^3 + ax + b$, with a and b in Z, Siegel (Lang, 1978) showed that the associated group has finitely many points with integer coordinates.

References

Cassels, J., 1991. London Mathematical Society Student Texts 24 Lectures on Elliptic Curves. Cambridge University Press.
Cayley, A., 1889. On the Intersection of Curves. Cambridge U. Press.
Dumber, S., 2019. Torsion on Elliptic Curves and Mazur's Theorem. U. Chicago. Edu.
Lang, S., 1978. Elliptic Curves, vol. 231. Springer, 3-540-08489-4.
Silverman, J., 1986. 0-387-96203-4 Arithmetic of Elliptic Curves, vol 106. Springer-Verlag.

APPENDIX

D

Young diagrams

There are several versions of Young diagrams, but almost all are constructed like bar graphs employing a finite number of cells or unit squares (Yong, 2007). These diagrams are important in representations of Fock space as well as quark flavor eigenstates (Bouttier, 2019; Sagan, 2001). The diagrams employed here are often located in the fourth quadrant and always involve integral translates of the unit square $(0, 1] \times [-1, 0)$. The structure forms contiguous horizontal bars all left justified, usually along the y-axis. The structure has a starting point at (p,q) in $Z \times Z$. This is the point at which the upper left-hand corner of the first square is located. Most often, this point is (0, 0). Horizontally, there are n1 contiguous cells forming a block or bar of length n1, located at $(p, p + n1] \times [q - 1, q)$ where n1 is greater than or equal to 1. Underneath, there can be another bar of length n2 less than or equal in size to n1. This bar is formed by n2 cells and is $(p, p + n2] \times [q - 2, q - 1)$. This pattern continues with nonincreasing nj cells, one underneath the other. Instead of drawings of Young diagrams, they are specified as (p,q), (n1, n2, ..., nk), the latter grouping is a finite number of nonnegative numbers nj, which is a nonincreasing sequence. The profile of a Young diagram is the boundary points made up of piecewise continuous horizontal and vertical line segments located on the right and lower parts of the structure. Also, the height *h*, for a Young diagram, is the number of nonzero nj values. In short, the Young diagram can be represented as a k tuple bound vector, $(n1, n2, \ldots, nk)_{(p,q)}$.

Example D.1:

Consider the Young diagram in Fig. D.1. It is given by (p, q) = (0, 0) and (n1, n2, n3, n4, 0, ...) = (5, 3, 2, 1, 0 ...). The profile for this structure consists of a path-connected set of line segments: It starts from the bottom leftmost corner and travels to the left horizontal line segment, $\{(x, y) \mid y = -4, x \text{ in } (0, 1]$. Then at $x = 1$, y in $[-4, -3)$, $y = -3$, x in [1, 2], $x = 2$, y in $[-3, -2)$, $y = -2$, x in [2, 3], $x = 3$, y in $[-2, -1)$, $y = -1$, x in [3, 4), $y = -1$, x in [4, 5], $x = 5$, y in $(-1, 0]$. Finally, the height for this structure is $h = 4$. Since $q = 0$, $-h = -4$ is the minimum value attained by this structure. Bound vector representation uniquely specifies this diagram: (5, 3, 2, 1) (0, 0), where the last integer pair specifies the

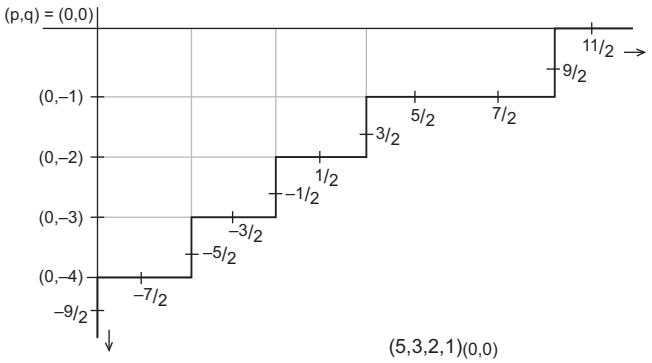


FIGURE D.1 Young diagram.

left-hand corner of the diagram and is usually written as a subscript. See Section 1.9, for an additional use of this representation. The Young diagram is marked starting on the $x = 0$ axis, half a unit under the bottom cell at $-9/2$. This is followed by adding one on each side of the cells in the profile. The markings are important in determining Maya diagrams. If the point (p, q) is arbitrary, an equivalence class of Young diagrams occurs. Moreover, the origin can be considered the coset leader in this case.#

References

Bouttier, J., *Fermions in combinatorics: random permutations and partitions*, fermions.sciencesconf.org. 2019.
Sagan, B., 2001. 0-387-95067-2 The Symmetric Group. Springer.
Yong, A., 2007. What is a Young tableau. Not. AMS 54 (2).

APPENDIX

E

Young diagrams and the symmetric group

The symmetric group Sn is important in group theory mainly because of Cayley's theorem. That is, the symmetric group, Sn, contains all subgroups isomorphic to any group of n elements or less. This is used in describing Fock space. It will be seen that the Young diagrams described in Appendix A.4 provide visual descriptions of irreducible representations of Sn. Of importance, besides elements of Sn being composed of products of transpositions of elements, it can be expressed as disjoint products of cyclic generators. Each cycle has a finite length, namely the number of elements in a cycle. This number is positive and less than or equal to n. Classes for Sn are specified by the n numbers w1, w2, ..., wn, where wj indicates the number of cycles for Sn with length j. Due to the disjointness of cycles, it follows that the sum $\sum_{j=1}^{n} \mathrm{j\ wj} = \mathrm{n}$ (Sagan, 2001).

Example E.1:

Consider the permutation group of symmetries S7, provided in the matrix-type diagram below. Use this diagram to trace out the path and thereby determine a specific cycle within the symmetric group. Start with the top numeral, x, to determine the numeral vertically underneath the numeral, x, and call it y. Then if y appears on top, repeat the previous step; do this until x appears again on top. The number of times this up plus down process is performed for a specific x indicates the length of the cycle. The distinct elements involved form a cyclic group:

$$|1\,2\,3\,4\,5\,6\,7|$$
$$|7\,6\,4\,5\,3\,2\,1|.$$

Beginning with 1, $1 \rightarrow 7 \rightarrow 1$, so this is a length two cycle. Next, for 2, $2 \rightarrow 6 \rightarrow 2$, this is also length two cycle. Finally, for 3, $3 \rightarrow 4 \rightarrow 5 \rightarrow 3$, this is a length three cycle. The group Sn can be written as the product of cycles: {1 7) (2 6) (3 4 5), where order does not matter. Classes for S7 are denoted by wk. Here, k is the length of the cycle, and wk

specifies how many cycles there are of length k. In this case, w1 = 0, w2 = 2, w3 = 1, and w4 = w5 = w6 = w7 = 0. This follows because there exist two length k = 2 cycles and one k = 3 length cycle. Moreover, the sum 2 · w2 + 3 · w3 = 2 · 2 + 3 · 1 = 7.#

A partition is denoted by p = (p1, p2, . . ., pn, . . .). It is any sum of a finite or infinite sequence of nonincreasing nonnegative numbers pj, that is, each pj less than or equal to p (j + 1). Each pj is called a part of p. The sum of the parts is | p |, called the weight. When the weight is n, then p is said to be a partition of n. Since singleton cycles w1 exist for some symmetric group Sn, there can be n of these cycles, because there are no larger cycles. Also, for n > 1, there are Sn with one two cycle w2, the rest being one cycle, and so on, considering all the possibilities. Finally, some Sn are described by a single cycle of n elements wn. Thus, for any Sn, all the possibilities for the partition of n involving classes are p1 = w1 + w2 + . . . + wn, p2 = w2 + w3 + . . . + wn, . . ., pn = wn. A pj is used whenever there exists a k greater than or equal to j, such that there exists a wk. These quantities can be arranged in order from the largest p1 to the smallest pn in rows on top of each other. Each row consists of pj one unit square cells resulting in a structure called a Young diagram. The number of rows equals h and is the number of nonzero pj. In this application, the Young diagram representation is left justified and starts at the origin in the fourth quadrant in the x, y plane.

Example E.2:

In Example E.1, it was seen that there were w2 = 2, two cycles, and one three cycle w3. Accordingly, the partition of n = 7 will be given, but only for wj with nonzero entries. Accordingly, for p1 = w2 + w3 = 2 + 1 = 3, p2 = w2 + w3 = 3, p3 = w3 = 1, and so the sum of these is n = 7. The partition is (p1, p2, p3, . . . p7) = (3, 3, 1, 0, 0, 0, 0,) = (3, 3, 1). The assosciated Young diagram has seven cells. It has three rows with the top row consisting of three cells. Underneath is the second row of three cells. Finally, the bottom row has one cell. All rows are left justified. As a bound vector, this structure is $(3\ 3\ 1)_{(0,0)}$ (see Fig. E.1).#

There is a bijection between irreducible representations of Sn and Young diagrams. The Young diagrams enable simple identification of products of irreducible representations. In order to better identify representations with these diagrams, a Young tableau is utilized.

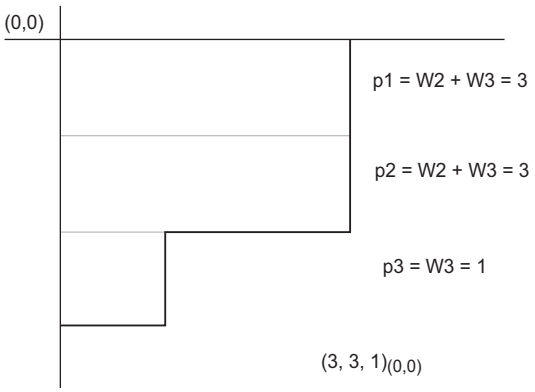


FIGURE E.1 Young diagram of permutation group.

This is a Young diagram with the n cells filled uniquely with integers 1, 2, . . ., n. For such a tableau, there exist n! ways of numbering the cells. When the numbering goes from left to right increasingly per row, continuing to the row below, then this is called a normal tableau. When the numbers only appear increasingly in a row or a column, this is called a standard Young tableau. A standard Young tableau arises from a normal tableau using a permutation.

References

Sagan, B., 2001. The Symmetric Group. Springer, 0-387-95067-2.

This page intentionally left blank

APPENDIX

F

Fundamental theorems in functional analysis

A listing of the basic theorems from functional analysis is provided below. Along with these theorems are several definitions relevant to the theorems proper (Apostle, 1974; Narici & Beckenstein, 2011; Rudin, 1973).

- Weierstrass approximation theorem

  Any continuous function f, defined on a compact interval [a, b] of R, can be uniformly approximated by a polynomial p, to within any ϵ, $\epsilon > 0$. Uniform approximation means that $\sup_x \mid f(x) - p(x) \mid < \epsilon$. A most elegant proof can be found using the Bernstein polynomials in Achieser (1956) and Narici (2004).
- Stone−Weierstrass theorem

  Let X be a locally compact Hausdorff space, and A a subalgebra of the Banach algebra $C_0(X)$, with the sup norm. Here, $C_0(X)$ consists of continuous real-valued functions, f(x) on X, which go to zero as $\mid x \mid \rightarrow \infty$. Then if it separates points and vanishes nowhere, then A is dense in C_0.

  A set S of functions f: A → B is said to separate points in A whenever there exists at least one function h in S that is injective, that is, it is 1−1. For all x, y in A, x is different from y, which implies that h(x) is different from h(y). The set of functions is said to vanish nowhere or at no point of A, which means that for every point x of A, there exists a function f in S such that f(x) is nonzero.

Example F.1:

The set of functions: $S_1 = \{f(x) = 1, g(x) = x^2\}$ does not separate points on the domain, A = [-2, 2], because at x = 1 and x = -1, f = g. However, $S_2 = \{f(x) = 1, g(x) = x^2, h(x) = x\}$ does separate points in A, since S_2 contains a function h, which is 1−1 on A.

Example F.2:

The set of functions: $S_1 = \{f(x) = 1, g(x) = x^2\}$ does vanish nowhere on the domain, A = [-2, 2]. However, $S_3 = \{g(x) = x^2, h(x) = x\}$ does not vanish nowhere on this domain, because at zero there is no function in this set that is nonzero.

- The Stone−Weierstrass theorem

  For any compact set K, let A be possibly a nonunital, associative self-adjoint algebra of continuous functions f: K → C, which separates points in K. If additionally, A vanishes at no point of K, or if A is unital, then A is dense in C(K) using the uniform norm.
- The Arzela−Ascoli theorem

  Let $\{f_n\}$, n = 0, 1, 2, ... be a sequence of real-valued functions on the compact set [a, b] in R. If this sequence is both uniformly bounded and equicontinuous, then there exists a subsequence $\{f_{n_k}\}$, which converges uniformly on [a, b]. When every subsequence of $\{f_n\}$ converges uniformly on [a, b], then $\{f_n\}$ is both uniformly bounded and equicontinuous.

  When there exists a constant $M > 0$, such that for all x in [a, b] and all nonnegative integers n, such that $| f_n | < M$, then the sequence of continuous functions f_n is said to be uniformly bounded.

  The sequence $\{f_n\}$ is uniformly equicontinuous whenever for $\epsilon > 0$, for all x, and y is in [a, b], and n nonnegative, there is a $\delta > 0$, such that if $| x - y | < \delta$, then $| f_n(x) - f_n(y) | < \epsilon$.
- Hahn−Banach theorem

  Let f be a continuous linear functional on a vector subspace M of a normed vector space X. In this case, f has a continuous linear extension F to all of X, and moreover, $|| f || = || F ||$. An excellent historical account appears in the study of Narici (2004).
- Open mapping theorem

  For an onto, continuous, linear operator T, T: A—-> B, where A and B are Banach spaces; then T is an open map. Thus, for any open set U in A, T(U) is an open set in B.
- Principle of uniform boundedness

  Let B(X,Y) be the space of all bounded operators from the Banach space X into the normed vector space Y, and let F be a subset of B(X,Y). If the $\sup_{TinF} || T(x) || < \infty$, for all x in X, then letting $c = \sup_{TinF} || T(x) ||$ when $|| x || = 1$, it follows that $c = \sup_{TinF} || T ||$.
- Bounded inverse theorem

  For X and Y Banach spaces if T: X → Y is bounded and invertible, then T^{-1} is also bounded.
- Baire category theorem

  A subset S of a topological space T is of the second category if S cannot be written as a countable union of nowhere dense sets in T. This means that at least one set within the union has closure with nonempty interior.
- Banach−Steinhaus theorem

  Consider X and Y both topological vector spaces, and $\Gamma = \{T_a\} ainR$, where T_a is a continuous linear map from X to Y. Let B denote the set of all points x in X whose orbits $\Gamma(x) = \{T_a(x)$, such that T_a is in $\Gamma\}$ are bounded in Y. If B is of the second category

in X, then B = X, and the collection of functions Γ is equicontinuous. An excellent presentation is made by Stover (2023).

- Closed graph theorem

For a linear operator T, T: A → B, where A and B are Banach spaces, then T is bounded iff the graph of T is closed. Equivalently, when given that $v_n \to v$, all in A and also given that T $v_n \to w$, all in B, if Tv = w, then T is bounded.

The Cartesian product, A × B is called the product space. It possesses the product topology, also called the Tychonoff topology or the coarsest topology. It is the topology containing the smallest number of open sets when using continuous projections, p_a and p_b, where p_a: A × B → A and p_b: A × B → B.

The graph of T is graph(T) = {(v, Tv) such that v is in domT}. The map T: A → B is closed in the graph A × B whenever it is a closed subset of the product space A × B with the Tychonoff topology. Equivalently, when given that $v_n \to v$, for all v_n and v in A, and also given that T $v_n \to w$, for all T v_n and w in B, if Tv = w, then T is bounded.

References

Achieser, N., 1956. 0−8044-4019-0 Theory of Approximation. Unger.

Apostle, T., 1974. Mathematical Analysis, 2ed. Pearson Pub Co.

Narici, L., On the Hahn-Banach theorem, Proceedings of the second international course of math analysis, Granada, 2004.

Narici, L., Beckenstein, E., 2011. 978−1584888666 Topological Vector Spaces, Pure and Applied Math, 2nd ed. CRC press.

Rudin, W., 1973. 0−07-054236-8 Functional Analysis. McGraw-HIll.

Stover, C., 2023. Banach-Steinhaus theorem, from MathWorld.wolfram.com.

This page intentionally left blank

APPENDIX

G

Sturm–Liouville differential equations and consequences

A Sturm–Liouville equation is a second-order linear differential equation (DE) involving functions p, q, and r as well as a parameter λ. This DE is given by $(p(x)y')' + (q(x) + \lambda\, r(x))\, y = 0$. The prime indicates a derivative. A Sturm–Liouville problem consists of the aforementioned DE along with a domain, for instance, the interval (a, b). It also contains boundary conditions at the endpoints for the closure of this interval. Moreover, it is assumed that p, p′, q, and r are continuous on the domain, where p′ denotes the derivative of p and the function $p > 0$, on this domain. The function r is often represented as w and is referred to as the weight. The parameter λ is not specified a priori. It is found as part of the nontrivial solution, and in this case, it is called an eigenvalue. The solutions to the DE are called eigenvectors or eigenfunctions corresponding to the eigenvalues.

The DE can be reconfigured into a self-adjoint operator, in a Hilbert space H of functions. The inner product in H is defined as follows: $< f, g > = \int_a^b f * g\, dx$. Principal references are Bochner (1929) and Birkhoff and Rota (1978). Solutions of the Sturm–Liouville DE exist under various boundary conditions, in particular, when they are referred to be separable. In this case, there exist real-valued scalars c_1, c_2, d_1, and d_2 such that $c_1^2 + c_2^2 > 0$, and $d_1^2 + d_2^2 > 0$, where the boundary conditions hold: $c_1\, y(a) + c_2\, y'(a) = 0$ and $d_1\, y(b) + d_2 y'(b) = 0$. The Sturm–Liouville DE along with the aforementioned boundary values (BVs) constitutes a system wherein the DE and BV have

1. Solutions, which consist of eigenfunctions $y_n(x)$, in the interval (a, b). They are unique up to a scalar multiple. The interval might extend to infinity in one or two directions. The corresponding eigenfunctions have n-1 zeros in this interval.
2. Eigenvalues; each eigenfunction has an associated unique real eigenvalue, and they are ordered $\lambda_1 < \lambda_2 < \lambda_3 < \ldots$
3. Weighted Hilbert space. Normalized eigenfunctions form an ON basis of functions in a weighted L^2, Hilbert space. The inner product, $< y_n, y_m >$, is given by the weighted integral: $\int_a^b y_n(x) y_m(x) u(x) dx$ = zero, except when $n = m$, then the integral equals one after normalization.

From a practical point of view, finding the weight u (x) is the most important criterion because it defines the inner product for the Hilbert space in L^2. In short, for the linear operator: L(y) = a y″ + b y′ + c y, the weighting function is $u = e^{\int (b-a')/a dx}$. This is illustrated below for several prominent polynomial classes. After developing all the weighting functions, the Hermite polynomials and the Laguerre polynomials are described in detail.

Example G.1:

Legendre polynomials: $L(y) = (1-x^2)\, y'' - 2x = \lambda\, y$. Then $u = e^{\int (b-a')/a dx} = e^0 = 1$. For these functions on the interval [− 1, 1], there is no need for boundary conditions to make L self-adjoint. The Legendre polynomials, y_n, form an ON basis, with an inner product $\int_{-1}^{1} y_n(x) y_m(x) dx$ = zero, except when $n = m$; then the integral can be normalized to one. The DE with the eigenvalues substituted in is $(1-x^2)\, y'' - 2x - n\,(n+1) = 0$.

Example G.2:

The Hermite polynomials: $H(y) = y'' - 2xy' = -\lambda\, y$. Then, $u = e^{\int (b-a')/a dx} = e^{\int (-2x) dx} = e^{-x^2}$. For these functions on $(-\infty, \infty)$, the Hermite polynomials form an ON basis $\int_{-\infty}^{\infty} y_n(x) y_m(x) e^{-x^2} dx$ = zero, except when $n = m$; then the integral can be normalized to one.

Example G.3:

The Laguerre polynomials: $L(y) = xy'' + (1 - x)\, y' = -\lambda\, y$ on $[0, \infty)$. Then $u = e^{\int (b-a')/a dx} = e^{\int (-1) dx} = e^{-x}$. They form an ON basis $\int_0^{\infty} y_n(x) y_m(x) e^{-x} dx$ = zero, except when $n = m$; then the integral can be normalized to one.

Example G.4:

Harmonic oscillator: $H(y) = 1/2\, (-h^2/m\, y'' + m(w_0\, x)^2)\, y = \lambda\, y$. Then the solutions form an ON set. Weight $u = e^{\int (b-a')/a dx} = e^0 = 1$. In this case, the eigen functions are in L^2. Accordingly, the weighting function is one in this case.

Example G.5:

The confluent hypergeometric equation: $L(y) = x\, y'' + (c - x)\, y' = \lambda\, y$. The weight is $u = e^{\int (b-a')/a dx} = e^{\int (c-x-1)/x dx} = |x|^{c-1} e^{-x}$. The DE is $x\, y'' + (c - x)\, y' + n\, y = 0$.

Example G.6:

Chebyshev polynomials: $L(y) = (1-x^2)\, y''$ - $x\, y' = \lambda\, y$, $L(y) = (1-x^2)\, y'' - 3\, x\, y' = \lambda\, y$. First, it is self-adjoint on [− 1, 1]. Weight $u = e^{\int (b-a')/a dx} = e^{\int (x)/(1-x^2) dx} = 1/(1-x^2)^{1/2}$. $\int_{-1}^{1} y_n(x) y_m(x)/(1-x^2)^{1/2}\, dx$ = zero, except when $n = m$; then the integral can be normalized to one. The DE with the eigenvalues substituted into the DE is $(1-x^2)\, y'' - x\, y' + n^2\, y = 0$.

Example G.7:

Jacobi polynomials: $L(y) = (1-x^2)\ y'' + (e\ x + g)\ y' = \lambda\ y$, on $[-1, 1]$. Weight $u = e^{\int (b-a')/a dx} = e^{\int (ex+g-2x)/(1-x^2) dx}$. For $g = -1, -2, -3$ and $e = 0$, the Jacobi polynomials become the Legendre and Chebyshev polynomials, respectively. If $-e < g < e$, then the DE is $(1-x^2)\ y'' + (e\ x + g)\ y' + [n\ (n-1) - n\ e]\ y = 0$.

Many classes of special functions can be defined or described in several different ways. For the Hermite polynomials, they satisfy the following:

1. Differential equation: $h'' - 2xh' + 2nh = 0$, or in terms of Sturm–Liouville differential equation, $e^{-x^2}\ h'' - 2xe\ h' + 2ne^{-x^2}\ h = 0$
2. Rodrigues formula: $hn = (-1)^n\ e^{x^2}\ d^n\ e^{-x^2}\ /\ dx^n$, $n = 0, 1, 2, \ldots$.
3. Difference equation: $h(n+1) - 2xhn + 2nh(n-1) = 0$
4. Generating function: $e^{(2xt-t^2)}$ times the sum $\sum (t^k\ /\ k!\ Hk)$, $k = 0, 1, 2, \ldots$

When they are normalized, they are symbolized by Hn, become an ON basis (not weighted and no longer polynomials) for L2 and are given by $Hn = e^{-1/2x^2} hn/(2^n n!(pi)^{1/2})^{1/2}$. These are called Hermite ON functions.

Also for the Laguerre polynomials, $y = Ln(x)$, they satisfy the following:

1. Differential equation: $xy'' + (1-x)\ y' + n\ y = 0$, x in [0, infinity), or in terms of Sturm–Liouville $(e^{-x}\ xy')' + n\ e^{-x}\ y = 0$.
2. Rodrigues formula: $Ln(x) = [(d/\ dx - 1)^n]\ x^n\ /\ n!$.
3. Difference equation: $L(n+1)(x) = [(2\ n + 1 - x)\ Ln(x) - n\ L(n-1)(x)]/(n+1)$, for $n = 1, 2, 3, 4, \ldots$
4. Generating function: Sum, $\sum t^n Ln(x) = (1-t)^{-1} e^{-tx/(1-t)}$, $n = 0, 1, 2, \ldots$.

A second-order linear homogenous DE can be represented as a Sturm–Liouville DE by defining an integrating factor p. The method is outlined below, more rigorously. For a linear operator, $L(y) = a\ y'' + b\ y' + c\ y$, if $p = e^{\int (b-a')/a dx}$ is finite over an interval I, and V is a vector space of functions such that

1. V is invariant under L.
2. For every y in V, the integral of $p\ y^2$ over I is finite.
3. For u and y in V, the difference of $p\ a\ (u'\ y - u\ y') = p\ a\ (u'\ y - u\ y')$ evaluated at the end points of I vanishes, then L is self-adjoint with respect to the inner product $<y, u> = \int p\ y\ u\ dx$.

References

Birkhoff, G., Rota, G.-C., 1978. Ordinary Differential Equations, 3rd ed. John Wiley and Sons.
Bochner, S., 1929. Uber Sturm-Liouville. Plansysteme. Math. Z. 29.

This page intentionally left blank

Index

Note: Page numbers followed by "*f*" refer to figures.

D

Q

R

S

Many-Sorted Algebras for Deep Learning and Quantum Technology

Charles R. Giardina

Many-Sorted Algebras for Deep Learning and Quantum Technology presents a precise and rigorous description of basic concepts in quantum technologies and how they relate to deep learning and quantum theory. Current merging of quantum theory and deep learning techniques provides the need for a source that gives readers insights into the algebraic underpinnings of these disciplines. Although analytical, topological, probabilistic, as well as geometrical concepts are employed in many of these areas, algebra exhibits the principal thread; hence, this thread is exposed using many-sorted algebras. This book includes hundreds of well-designed examples that illustrate the intriguing concepts in quantum systems. Along with these examples are numerous visual displays. In particular, the polyadic graph shows the types or sorts of objects used in quantum or deep learning. It also illustrates all the inter and intra-sort operations needed in describing algebras. In brief, it provides the closure conditions. Throughout the book, all laws or equational identities needed in specifying an algebraic structure are precisely described.

Key features

- Includes hundreds of well-designed examples to illustrate the intriguing concepts in quantum systems
- Provides precise description of all laws or equational identities that are needed in specifying an algebraic structure
- Illustrates all the inter and intra sort operations needed in describing algebras

About the author

Charles R. Giardina was formerly with Bell Telephone Laboratories, Whippany, NJ, United States. His research interests include digital signal and image processing, pattern recognition, artificial intelligence, and the constructive theory of functions. Dr. Giardina has authored numerous papers in these areas and several books including *Mathematical Models for Artificial Intelligence and Autonomous Systems*, Prentice Hall; *Matrix Structure Image Processing*, Prentice Hall; *Parallel Digital Signal Processing: A Unified Signal Algebra Approach*, Regency; *Morphological Methods in Image and Signal Processing*, Prentice Hall; *Image Processing – Continuous to Discrete: Geometric, Transform, and Statistical Methods*, Prentice Hall; and *A Unified Signal Algebra Approach to Two-Dimensional Parallel Digital Signal Processing*, Chapman and Hall/CRC Press.

ISBN 978-0-443-13697-9

9 780443 136979

ELSEVIER

MK

MORGAN KAUFMANN PUBLISHERS

An imprint of Elsevier

elsevier.com/books-and-journals