Sample chapter: prepublication material

Sample chapter - prepublication draft of a planned textbook; not the complete book. Content may change before publication.

Copyright (c) 2026 Sajid Muhaimin Choudhury. All rights reserved.

No licence to reproduce, adapt or redistribute this chapter is granted by its publication on the author's website.

EEE 6516 - Quantum Computing

Lecture Notes

Chapter 1

Why Quantum Computing? The Mathematical Toolkit

Sajid Muhaimin Choudhury

Copyright © 2026 Sajid Muhaimin Choudhury. All rights reserved.

This manuscript is unpublished work in progress. No part of it may be reproduced, distributed, published, modified, sold or used to create derivative works without the prior written permission of the copyright holder, except as permitted by applicable law.

Third-party names and trademarks mentioned in this manuscript remain the property of their respective owners.

Student edition • Phase 0 setup • Build `bc1f8c2` • Typeset 11th October 2026

Contents

1 Why Quantum Computing? The Mathematical Toolkit **1**

1.1 Why quantum computing? 1
1.2 From bits to qubits 3
1.3 Complex numbers revisited 4
1.4 Kets, bras and normalisation 5
1.5 Inner products and bases 6
1.6 Operators as matrices 8
1.7 Hermitian and unitary matrices 9
1.8 Eigenvalues and the spectral theorem 10
1.9 A first look at the tensor product 13
1.10 Setting up the simulators 13
1.11 More linear algebra: trace, commutators, Gram–Schmidt 14
1.12 Vector spaces in general 15
1.13 Summary 15
1.14 Multiple-choice questions 16
1.15 Problems 17

References **19**

© 2026 S. M. Choudhury • Working draft • Not for redistribution

Chapter 1

Why Quantum Computing? The Mathematical Toolkit

A quantum computer is described by linear algebra over the complex numbers. Nothing in this course requires more mathematics than an electrical engineer has already met in circuit theory and signals: complex numbers, vectors, matrices and eigenvalues. What is new is the way these familiar objects are read. A list of complex numbers will stand for the state of a machine, a matrix for an operation on it, and an inner product for the probability of an experimental outcome. This chapter explains why such a machine is worth building and then assembles the toolkit, one idea at a time, with a short simulation for each.

1.1 Why quantum computing?

A classical register of n bits is, at any moment, in exactly one of its 2^n configurations, and n numbers suffice to describe it. A register of n qubits is different. As Chapter 2 will establish, its complete description is a list of 2^n complex numbers, one for every configuration, called *amplitudes*. Adding one qubit doubles the length of the list.

This growth has a direct engineering consequence. To simulate n qubits on an ordinary computer we must store all 2^n amplitudes.

Worked Example 1.1 – The memory wall

A simulator stores each amplitude as a complex number made of two 64-bit floating-point numbers (16 B). How much memory does it need to hold the state of 30 qubits? Of 50 qubits?

Solution. The state of n qubits has 2^n amplitudes, so it needs $16 \cdot 2^n$ bytes $= 2^{n+4}$ bytes.

1. $n = 30$: 2^{34} bytes $= 2^4 \cdot 2^{30}$ bytes $= 16$ GiB, the main memory of a well-equipped laptop.
2. $n = 50$: 2^{54} bytes $= 16$ PiB $\approx 1.8 \times 10^{16}$ bytes, far beyond any single machine.

Each additional qubit doubles the requirement; twenty more qubits multiply it by about a million.

Simulation 1.1 – The memory wall in numbers

The same arithmetic for $n = 10, 20, \ldots, 50$, with binary prefixes (1 KiB = 1024 B).

© 2026 S. M. Choudhury • Working draft • Not for redistribution

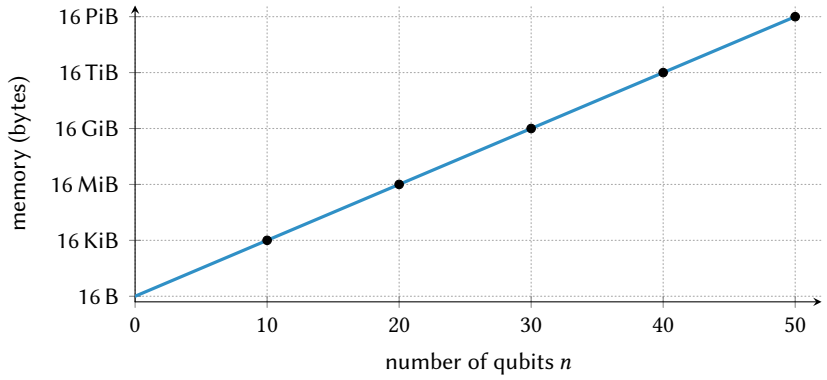


Figure 1.1 Memory needed to store the state vector of n qubits at 16 B per amplitude. The marks are the values printed in Simulation 1.1.

```
BYTES = 16      # one complex number = two 64-bit floats
UNITS = ["B", "KiB", "MiB", "GiB", "TiB", "PiB", "EiB"]


def human(nbytes):
    k = 0
    while nbytes >= 1024 and k < len(UNITS) - 1:
        nbytes /= 1024
        k += 1
    return f"{nbytes:g} {UNITS[k]}"


for n in range(10, 51, 10):
    amps = 2 ** n                   # amplitudes of n qubits
    size = human(BYTES * amps)
    print(f"n = {n}: {amps:>21,d} amplitudes, {size}")
```

Output.

```
n = 10:                 1,024 amplitudes, 16 KiB
n = 20:             1,048,576 amplitudes, 16 MiB
n = 30:         1,073,741,824 amplitudes, 16 GiB
n = 40:     1,099,511,627,776 amplitudes, 16 TiB
n = 50: 1,125,899,906,842,624 amplitudes, 16 PiB
```

What to observe. Every ten qubits multiply the memory by $2^{10} = 1024$; Figure 1.1 plots the whole curve.

The memory wall is the reason Feynman proposed, in a talk given in 1981 and published the following year, that quantum systems should be simulated by machines that are themselves quantum [2]. It is also the reason this course can simulate only small examples: every simulation in these notes uses at most a handful of qubits.

What a quantum computer is good for

A long list of amplitudes does not make a quantum computer a faster classical computer. Three kinds of problem are known for which a quantum algorithm beats

© 2026 S. M. Choudhury • Working draft • Not for redistribution

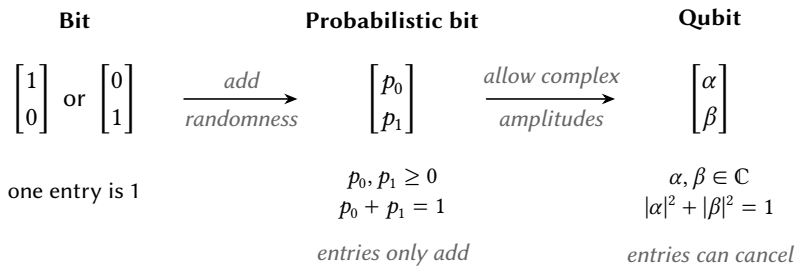


Figure 1.2 A bit, a probabilistic bit and a qubit, each described by a two-entry vector. Only the rules for the entries change.

the best classical method that we know of [4, ch. 1]:

- *factoring and discrete logarithms* (Shor's algorithm, Chapter 6): polynomial time, whereas every known classical algorithm takes super-polynomial time;
- *unstructured search* (Grover's algorithm, Chapter 5): about $\sqrt{N}$ steps instead of about N, a quadratic gain;
- *simulation of quantum systems* such as molecules and materials, Feynman's original motivation.

For most everyday computing no quantum advantage is known, and none is expected. Today's devices are also small and noisy: Preskill coined the term *noisy intermediate-scale quantum* (NISQ) for machines whose gates are too error-prone for long computations [6]. Making qubits reliable enough for large algorithms is the subject of Chapter 8; the hardware itself is surveyed, with dated figures, in Chapter 9.

Common mistake ('A quantum computer tries every answer at once'). It is tempting to read a list of 2^n amplitudes as 2^n computations running in parallel, with the best answer read off at the end. That is not how it works. Reading out n qubits yields only n bits (Chapter 2), and a careless readout returns a random configuration. A quantum algorithm succeeds only if it arranges the amplitudes so that wrong answers cancel and the right answer becomes likely: the cancellation of Section 1.2, used deliberately.

1.2 From bits to qubits

Three kinds of one-bit register, compared in Figure 1.2, isolate what is genuinely new about a qubit.

A bit. A deterministic bit is 0 or 1. Writing 0 as the vector $(1,0)^{\mathsf{T}}$ and 1 as $(0,1)^{\mathsf{T}}$ looks like a complication, but it lets us treat the next two cases in the same way.

A probabilistic bit. A bit produced by a fair coin is 0 with probability p_0 and 1 with probability p_1. Its description is the *probability vector* $(p_0, p_1)^{\mathsf{T}}$, with non-negative entries that add up to one. A random operation is a matrix whose columns are probability vectors. A 'random flip', which outputs 0 or 1 with equal probability

© 2026 S. M. Choudhury • Working draft • Not for redistribution

whatever the input, is

$$F = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \qquad F\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \qquad F\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}. \tag{1.1}$$

Two random flips leave the bit as random as one. Probabilities only ever add, so once randomness is introduced, no further random operation can remove it.

A qubit. A qubit is described by a vector $(\alpha, \beta)^{\mathsf{T}}$ whose entries are *complex* numbers with $|\alpha|^2 + |\beta|^2 = 1$; Chapter 2 explains why $|\alpha|^2$ and $|\beta|^2$ behave as probabilities. An operation is a matrix that preserves this condition. A quantum analogue of the flip is the matrix

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad H\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \qquad H\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} + \frac{1}{2} \\ \frac{1}{2} - \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{1.2}$$

After one application both entries have the same size, just as for the random flip. After a second application the entries of the second component *cancel* and the register is back in its starting state. Negative (more generally, complex) entries make this possible, and this cancellation is what every quantum algorithm exploits.

1.3 Complex numbers revisited

Amplitudes are complex numbers, and an electrical engineer already handles them daily as phasors. One change of symbol: this course writes the imaginary unit as i, as physics and computing texts do, where circuit texts write j.

A complex number $z = a + \mathrm{i}b$ has a *real part* a, an *imaginary part* b, a *modulus* $|z| = \sqrt{a^2 + b^2}$ and a *phase* (argument) ϕ, so that

$$z = |z|\,\mathrm{e}^{\mathrm{i}\phi} = |z|(\cos\phi + \mathrm{i}\sin\phi). \tag{1.3}$$

Its *complex conjugate* is $z^* = a - \mathrm{i}b = |z|\mathrm{e}^{-\mathrm{i}\phi}$, and the identity used most often in this course is

$$z^* z = |z|^2 = a^2 + b^2. \tag{1.4}$$

Multiplying by $\mathrm{e}^{\mathrm{i}\theta}$ rotates z by θ without changing its modulus (Figure 1.3).

> **Design insight (Phasors and amplitudes).** A phasor $V = |V|\mathrm{e}^{\mathrm{i}\phi}$ encodes the amplitude and phase of a sinusoidal voltage, and adding phasors predicts how two sinusoids interfere. An amplitude behaves the same way under addition, so two contributions can reinforce or cancel. The analogy stops at interpretation: a phasor describes a deterministic waveform, whereas the squared modulus of an amplitude is a probability.

Checkpoint. For $z = 1 + \mathrm{i}$ and $w = \sqrt{3} - \mathrm{i}$, find $|zw|$ and the phase of zw without multiplying out.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

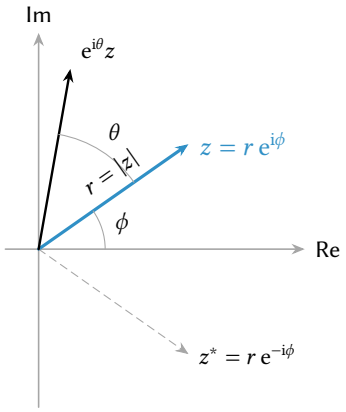


Figure 1.3 An amplitude drawn as a phasor: modulus r, phase ϕ, its conjugate, and the effect of multiplying by $e^{i\theta}$.

1.4 Kets, bras and normalisation

From now on the vectors describing a qubit are written in *Dirac notation*. The notation is only a compact way of naming column and row vectors, but it makes the algebra of the coming chapters much shorter.

Definition 1.1 (Ket and computational basis). A *ket* $|\psi\rangle$ is a column vector of complex numbers. For one qubit the two kets

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1.5}$$

form the *computational basis*, and every two-entry column vector can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = (\alpha, \beta)^{\mathsf{T}}$.

Definition 1.2 (Bra). The *bra* $\langle\psi|$ is the conjugate transpose of $|\psi\rangle$: a row vector whose entries are the complex conjugates of those of the ket,

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad \Longrightarrow \quad \langle\psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}. \tag{1.6}$$

Definition 1.3 (Normalised state). A ket $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is *normalised* if $|\alpha|^2 + |\beta|^2 = 1$. Only normalised kets describe qubits; any non-zero ket can be normalised by dividing it by $\sqrt{|\alpha|^2 + |\beta|^2}$.

Worked Example 1.2 – Normalising a state

Normalise $|\chi\rangle = (1 + \mathrm{i})|0\rangle + 2|1\rangle$, and give $|\alpha|^2$ and $|\beta|^2$ for the result.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

Solution. The squared moduli are $|1+\mathrm{i}|^2 = 1^2+1^2 = 2$ and $|2|^2 = 4$, so the norm is $\sqrt{6}$ and

$$|\psi\rangle = \frac{1+\mathrm{i}}{\sqrt{6}}|0\rangle + \frac{2}{\sqrt{6}}|1\rangle, \qquad |\alpha|^2 = \tfrac{1}{3}, \quad |\beta|^2 = \tfrac{2}{3}.$$

Common mistake (Squaring instead of taking the squared modulus). Normalisation uses $|\alpha|^2 = \alpha^*\alpha$, never α^2. In Worked Example 1.2, $(1+\mathrm{i})^2 + 2^2 = 4 + 2\mathrm{i}$ is not even a real number. The same slip appears when forming a bra: forgetting to conjugate turns $\langle\psi|$ into a plain transpose.

1.5 Inner products and bases

Definition 1.4 (Inner product, norm, orthogonality). The *inner product* of $|\phi\rangle = (\phi_0, \phi_1)^{\mathsf{T}}$ and $|\psi\rangle = (\psi_0, \psi_1)^{\mathsf{T}}$ is the bra of the first times the ket of the second,

$$\langle\phi\,|\,\psi\rangle = \phi_0^{\,*}\psi_0 + \phi_1^{\,*}\psi_1. \tag{1.7}$$

The *norm* of $|\psi\rangle$ is $\|\psi\| = \sqrt{\langle\psi\,|\,\psi\rangle}$; two kets are *orthogonal* if $\langle\phi\,|\,\psi\rangle = 0$. A set of kets is an *orthonormal basis* if every ket in it has norm 1, any two of them are orthogonal, and every ket of the space is a combination of them.

The conjugate makes $\langle\psi\,|\,\psi\rangle = |\psi_0|^2 + |\psi_1|^2$ real and non-negative, so a normalised ket has norm 1. It also makes the order matter: swapping the two kets conjugates the result, $\langle\psi\,|\,\phi\rangle = \langle\phi\,|\,\psi\rangle^*$.

Besides the computational basis, one other basis appears constantly:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{1.8}$$

Worked Example 1.3 – The ± basis is orthonormal

Show that $\langle+\,|\,-\rangle = 0$ and $\langle+\,|\,+\rangle = 1$, and compute $\langle 0\,|\,+\rangle$.

Solution. The bra of $|+\rangle$ is $\frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)$ (the coefficients are real, so conjugation changes nothing). Using $\langle 0\,|\,0\rangle = \langle 1\,|\,1\rangle = 1$ and $\langle 0\,|\,1\rangle = \langle 1\,|\,0\rangle = 0$,

$$\langle+\,|\,-\rangle = \tfrac{1}{2}\big(\langle 0\,|\,0\rangle - \langle 0\,|\,1\rangle + \langle 1\,|\,0\rangle - \langle 1\,|\,1\rangle\big) = \tfrac{1}{2}(1 - 0 + 0 - 1) = 0,$$

and similarly $\langle+\,|\,+\rangle = \frac{1}{2}(1+1) = 1$. Finally $\langle 0\,|\,+\rangle = \frac{1}{\sqrt{2}}\langle 0\,|\,0\rangle = \frac{1}{\sqrt{2}} \approx 0.7071$.

Expanding in a basis. If $\{|e_0\rangle, |e_1\rangle\}$ is an orthonormal basis, the coefficients of $|\psi\rangle$ in that basis are inner products:

$$|\psi\rangle = c_0|e_0\rangle + c_1|e_1\rangle, \qquad c_i = \langle e_i\,|\,\psi\rangle. \tag{1.9}$$

(Take the inner product of both sides with $\langle e_i|$; orthonormality removes the other term.) For real amplitudes this is ordinary projection onto perpendicular axes, as Figure 1.4 shows.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

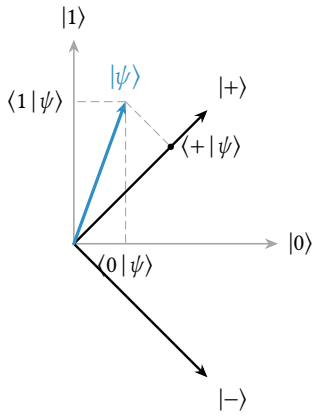


Figure 1.4 One state and its coefficients in two orthonormal bases. The picture is exact only for real amplitudes; with complex amplitudes the geometry no longer fits in a plane.

Design insight (Inner products as signal projections). Fourier analysis expands a periodic signal in orthogonal basis functions, and each coefficient is an inner product of the signal with one basis function. Equation (1.9) is the same idea in two dimensions. Two differences matter: here the space is finite-dimensional, and the conjugate always sits on the bra, so the order of the two kets cannot be swapped freely.

Worked Example 1.4 – Changing basis

Express $|0\rangle$, and then $|\psi\rangle = \frac{1}{5}(3|0\rangle + 4\mathrm{i}|1\rangle)$, in the basis $\{|+\rangle, |-\rangle\}$.

Solution. By Equation (1.9), $\langle + \,|\, 0\rangle = \langle - \,|\, 0\rangle = \frac{1}{\sqrt{2}}$, so

$$|0\rangle = \tfrac{1}{\sqrt{2}}|+\rangle + \tfrac{1}{\sqrt{2}}|-\rangle.$$

The coefficients are $1/\sqrt{2} \approx 0.707$, not $1/2$: it is their squared moduli that equal $1/2$. For a general state $\alpha|0\rangle + \beta|1\rangle$ the same calculation gives $\frac{\alpha+\beta}{\sqrt{2}}|+\rangle + \frac{\alpha-\beta}{\sqrt{2}}|-\rangle$. For $|\psi\rangle$, $\alpha = \frac{3}{5}$ and $\beta = \frac{4\mathrm{i}}{5}$, so

$$\langle + \,|\, \psi\rangle = \frac{3 + 4\mathrm{i}}{5\sqrt{2}} \approx 0.4243 + 0.5657\mathrm{i}, \qquad \langle - \,|\, \psi\rangle = \frac{3 - 4\mathrm{i}}{5\sqrt{2}}.$$

Simulation 1.2 – Kets and inner products in NumPy

A ket is a one-dimensional complex array; the bra is its conjugate; the inner product is a sum of products.

```
import numpy as np

ket0 = np.array([1, 0], dtype=complex)
ket1 = np.array([0, 1], dtype=complex)
plus = (ket0 + ket1) / np.sqrt(2)
minus = (ket0 - ket1) / np.sqrt(2)
```

© 2026 S. M. Choudhury • Working draft • Not for redistribution

```python
def bra(ket):
    return ket.conj()          # entries of the row vector <ket|


def inner(phi, psi):
    return bra(phi) @ psi      # sum of conj(phi_i) psi_i


psi = np.array([3, 4j]) / 5    # (3|0> + 4i|1>)/5
print("<psi|psi> =", round(inner(psi, psi).real, 4))
print("<+|->     =", round(abs(inner(plus, minus)), 4))
print("<0|+>     =", round(inner(ket0, plus).real, 4))
print("<+|psi>   =", np.round(inner(plus, psi), 4))
print("<psi|+>   =", np.round(inner(psi, plus), 4))
print("np.vdot agrees:", np.isclose(np.vdot(psi, plus),
                                    inner(psi, plus)))
```

Output.

```
<psi|psi> = 1.0
<+|->     = 0.0
<0|+>     = 0.7071
<+|psi>   = (0.4243+0.5657j)
<psi|+>   = (0.4243-0.5657j)
np.vdot agrees: True
```

What to observe. $|\psi\rangle$ is normalised; $|+\rangle$ and $|-\rangle$ are orthogonal; $\langle + | \psi \rangle$ agrees with Worked Example 1.4, and swapping the order conjugates it. NumPy's built-in `np.vdot` conjugates its *first* argument, exactly like a bra.

1.6 Operators as matrices

An operation on a qubit is a linear map from kets to kets, represented by a 2×2 matrix. Two ways of looking at a matrix recur throughout the course.

Matrix elements. The entry in row i and column j of A is the number $\langle i | A | j \rangle$: the ith component of the image of the jth basis ket. For example, the matrix $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ has $\langle 0 | X | 1 \rangle = 1$ because it maps $|1\rangle$ to $|0\rangle$.

Outer products. A ket times a bra, $|\phi\rangle\langle\chi|$, is a matrix (a column times a row). Every matrix is a sum of such terms:

$$A = \sum_{i,j} \langle i | A | j \rangle \, |i\rangle\langle j|, \qquad \text{e.g.} \qquad X = |0\rangle\langle 1| + |1\rangle\langle 0|. \tag{1.10}$$

Definition 1.5 (Projector and completeness). For a normalised $|\psi\rangle$, the matrix $P = |\psi\rangle\langle\psi|$ is the *projector* onto $|\psi\rangle$. It satisfies $P^2 = P$ and $P^\dagger = P$ (Section 1.7), and $P|\phi\rangle = \langle\psi | \phi\rangle |\psi\rangle$ keeps only the component of $|\phi\rangle$ along $|\psi\rangle$.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

For any orthonormal basis $\{|e_i\rangle\}$ the projectors add up to the identity:

$$\sum_i |e_i\rangle\langle e_i| = I \qquad \text{(completeness relation).} \tag{1.11}$$

Completeness is Equation (1.9) written as a matrix identity: inserting $I = |e_0\rangle\langle e_0| + |e_1\rangle\langle e_1|$ in front of $|\psi\rangle$ produces the expansion in that basis.

1.7 Hermitian and unitary matrices

Definition 1.6 (Adjoint, Hermitian, unitary). The *adjoint* $A^\dagger$ of a matrix A is its conjugate transpose, $(A^\dagger)_{ij} = A_{ji}{}^*$. A matrix is

- *Hermitian* if $A^\dagger = A$;
- *unitary* if $U^\dagger U = I$, that is, $U^{-1} = U^\dagger$.

The bra of $A|\psi\rangle$ is $\langle\psi|A^\dagger$; this rule, applied to a product, gives $(AB)^\dagger = B^\dagger A^\dagger$. Hermitian matrices will describe measurable quantities and unitary matrices will describe the operations a quantum computer performs (Chapter 2). The two classes overlap but are not the same.

Theorem 1.7 (Unitaries preserve inner products). *If U is unitary, then $\langle\phi|U^\dagger U|\psi\rangle = \langle\phi|\psi\rangle$ for all kets; in particular $\|U\psi\| = \|\psi\|$, so a unitary maps normalised states to normalised states.*

Proof. The bra of $U|\phi\rangle$ is $\langle\phi|U^\dagger$, so the inner product of $U|\phi\rangle$ and $U|\psi\rangle$ is $\langle\phi|U^\dagger U|\psi\rangle = \langle\phi|I|\psi\rangle = \langle\phi|\psi\rangle$. □

Design insight (A lossless network is unitary). In microwave network theory, a two-port is described by a scattering matrix S that maps incident waves (a_1, a_2) to outgoing waves (b_1, b_2). With the waves normalised so that $|a_i|^2$ is a power, a lossless network satisfies $S^\dagger S = I$: power in equals power out [5]. A single-qubit unitary conserves $|\alpha|^2 + |\beta|^2$ in the same way (Figure 1.5). The analogy is mathematical: the network conserves the power carried by two physical waves, the unitary conserves the total probability of one quantum system.

Worked Example 1.5 – Hermitian, unitary, both or neither?

Classify $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $H = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, $S = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ and $N = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$.

Solution.

- X and H are real and symmetric, hence Hermitian; $X^2 = H^2 = I$, so $X^\dagger X = H^\dagger H = I$: *both.*
- $S^\dagger = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -\mathrm{i} \end{smallmatrix}\right] \neq S$, but $S^\dagger S = I$: *unitary only.*
- $B^\dagger = B$, but $B^\dagger B = 2B \neq I$: *Hermitian only.*

© 2026 S. M. Choudhury • Working draft • Not for redistribution

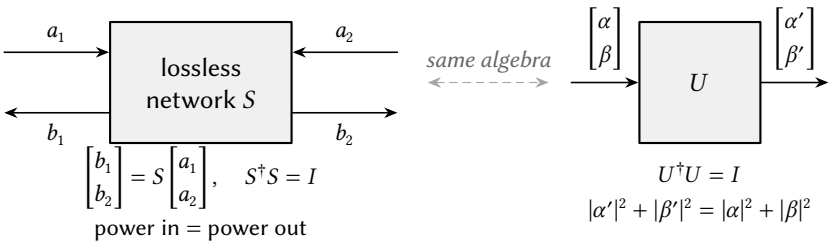


Figure 1.5 A lossless two-port and a single-qubit unitary obey the same algebra: both matrices satisfy $M^\dagger M = I$.

- $N^\dagger \neq N$ and $N^\dagger N = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right] \neq I$: *neither.*

Real entries, symmetry or non-negative entries do not make a matrix unitary; only $U^\dagger U = I$ does.

1.8 Eigenvalues and the spectral theorem

A non-zero ket $|v\rangle$ is an *eigenvector* of A with *eigenvalue* λ if $A|v\rangle = \lambda|v\rangle$. For Hermitian and unitary matrices the eigenvalues and eigenvectors are unusually well behaved, and that behaviour carries much of quantum mechanics.

Theorem 1.8 (Hermitian eigenvalues and eigenvectors). *The eigenvalues of a Hermitian matrix are real, and eigenvectors belonging to different eigenvalues are orthogonal.*

Proof. Let $A|v\rangle = \lambda|v\rangle$ with $|v\rangle \neq 0$. Then $\langle v \,|\, A \,|\, v\rangle = \lambda\langle v \,|\, v\rangle$. Taking the conjugate and using $A^\dagger = A$ gives $\langle v \,|\, A \,|\, v\rangle = \lambda^*\langle v \,|\, v\rangle$, so $\lambda = \lambda^*$. If also $A|w\rangle = \mu|w\rangle$ with $\mu \neq \lambda$, then $\langle w \,|\, A \,|\, v\rangle = \lambda\langle w \,|\, v\rangle$, and, applying A to the bra instead, $\langle w \,|\, A \,|\, v\rangle = \mu\langle w \,|\, v\rangle$ (μ is real). Subtracting, $(\lambda - \mu)\langle w \,|\, v\rangle = 0$, so $\langle w \,|\, v\rangle = 0$. □

Theorem 1.9 (Spectral theorem). *Every Hermitian matrix A has an orthonormal basis of eigenvectors $\{|v_i\rangle\}$ with real eigenvalues λ_i, and*

$$A = \sum_i \lambda_i |v_i\rangle\langle v_i|. \tag{1.12}$$

The orthogonality in Theorem 1.8 is the heart of the theorem; the existence of a full basis of eigenvectors is proved in [4, Box 2.2] (see also Section 1.12).

Theorem 1.10 (Unitary eigenvalues). *Every eigenvalue of a unitary matrix has modulus 1, $\lambda = e^{i\phi}$.*

Proof. If $U|v\rangle = \lambda|v\rangle$, then by Theorem 1.7 $\langle v \,|\, v\rangle = \|Uv\|^2 = |\lambda|^2\langle v \,|\, v\rangle$, so $|\lambda| = 1$. □

Worked Example 1.6 – Spectral decomposition of X

© 2026 S. M. Choudhury • Working draft • Not for redistribution

Find the eigenvalues and eigenvectors of X, write its spectral decomposition, and use it to compute $\mathrm{e}^{-\mathrm{i}\theta X/2}$.

Solution. From $X|+\rangle = |+\rangle$ and $X|-\rangle = -|-\rangle$ the eigenvalues are ± 1 with eigenvectors $|+\rangle$, $|-\rangle$ (orthonormal, by Worked Example 1.3). Hence $X = |+\rangle\langle+| - |-\rangle\langle-|$. A function of a Hermitian matrix acts on its eigenvalues (Definition 1.11 below), so

$$\mathrm{e}^{-\mathrm{i}\theta X/2} = \mathrm{e}^{-\mathrm{i}\theta/2}|+\rangle\langle+| + \mathrm{e}^{\mathrm{i}\theta/2}|-\rangle\langle-|.$$

Writing $\mathrm{e}^{\mp\mathrm{i}\theta/2} = \cos\frac{\theta}{2} \mp \mathrm{i}\sin\frac{\theta}{2}$ and using $|+\rangle\langle+| + |-\rangle\langle-| = I$ and $|+\rangle\langle+| - |-\rangle\langle-| = X$,

$$\mathrm{e}^{-\mathrm{i}\theta X/2} = \cos\frac{\theta}{2}\, I - \mathrm{i}\sin\frac{\theta}{2}\, X.$$

For $\theta = \pi/3$: $\cos\frac{\pi}{6} \approx 0.866$ on the diagonal and $-\mathrm{i}\sin\frac{\pi}{6} = -0.5\mathrm{i}$ off it.

Definition 1.11 (Function of a Hermitian matrix). If $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$, then $f(A) \coloneqq \sum_i f(\lambda_i)|v_i\rangle\langle v_i|$. For $f(x) = \mathrm{e}^x$ this agrees with the power series $\mathrm{e}^A = I + A + A^2/2! + \dots$.

An engineer meets the matrix exponential when solving the state equation $\dot{\mathbf{x}} = A\mathbf{x}$, whose solution is $\mathbf{x}(t) = \mathrm{e}^{At}\mathbf{x}(0)$. In Chapter 2 the evolution of a qubit has exactly this form, with A equal to $-\mathrm{i}/\hbar$ times a Hermitian matrix; e^{At} is then unitary, so probability is conserved.

Simulation 1.3 – Hermitian, unitary and spectral checks

The classification of Worked Example 1.5 and the spectral decomposition of $A = \left[\begin{smallmatrix} 1 & 1-\mathrm{i} \\ 1+\mathrm{i} & 2 \end{smallmatrix}\right]$, computed with `np.linalg.eigh`, the routine for Hermitian matrices.

```
import numpy as np

X = np.array([[0, 1], [1, 0]])
Y = np.array([[0, -1j], [1j, 0]])
H = np.array([[1, 1], [1, -1]]) / np.sqrt(2)
B = np.array([[1, 1], [1, 1]])
I = np.eye(2)


def dag(M):
    return M.conj().T          # adjoint: conjugate transpose


for name, M in [("X", X), ("Y", Y), ("H", H), ("B", B)]:
    herm = np.allclose(M, dag(M))
    unit = np.allclose(dag(M) @ M, I)
    print(f"{name}: Hermitian {herm!s:5}  unitary {unit}")

A = np.array([[1, 1 - 1j], [1 + 1j, 2]])
lam, V = np.linalg.eigh(A)     # columns of V are eigenvectors
print("eigenvalues of A:", np.round(lam, 6) + 0.0)
A2 = sum(l * np.outer(v, v.conj()) for l, v in zip(lam, V.T))
print("A = sum lam |v><v| :", np.allclose(A, A2))
P = sum(np.outer(v, v.conj()) for v in V.T)
```

© 2026 S. M. Choudhury • Working draft • Not for redistribution

```
print("sum |v><v| = I     :", np.allclose(P, I))
print("<v1|v2> = 0        :", np.isclose(np.vdot(*V.T), 0))
```

Output.

```
X: Hermitian True   unitary True
Y: Hermitian True   unitary True
H: Hermitian True   unitary True
B: Hermitian True   unitary False
eigenvalues of A: [0. 3.]
A = sum lam |v><v| : True
sum |v><v| = I     : True
<v1|v2> = 0        : True
```

What to observe. A has eigenvalues 0 and 3 (trace 3, determinant 0); its eigenvectors are orthogonal, rebuild A through Equation (1.12), and their projectors satisfy the completeness relation (1.11).

Simulation 1.4 – Three routes to $e^{-i\theta X/2}$

A numerical matrix exponential, the spectral decomposition and the closed form of Worked Example 1.6 must agree.

```
import numpy as np
from scipy.linalg import expm

X = np.array([[0, 1], [1, 0]], dtype=complex)
I = np.eye(2)
theta = np.pi / 3

U1 = expm(-1j * theta * X / 2)       # numerical exponential
lam, V = np.linalg.eigh(X)           # spectral decomposition
U2 = sum(np.exp(-1j * theta * l / 2) * np.outer(v, v.conj())
         for l, v in zip(lam, V.T))
U3 = np.cos(theta / 2) * I - 1j * np.sin(theta / 2) * X

print(np.round(U3, 4))
print("expm = spectral:", np.allclose(U1, U2))
print("expm = cos/sin :", np.allclose(U1, U3))
print("unitary        :", np.allclose(U3.conj().T @ U3, I))
```

Output.

```
[[0.866+0.j  0.   -0.5j]
 [0.   -0.5j 0.866+0.j ]]
expm = spectral: True
expm = cos/sin : True
unitary        : True
```

What to observe. All three agree and the result is unitary. This matrix is the rotation gate $R_x(\pi/3)$ of Chapter 2.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

1.9 A first look at the tensor product

Two qubits are described by a four-entry vector. If the first qubit is in $(a, b)^{\mathsf{T}}$ and the second in $(c, d)^{\mathsf{T}}$, the pair is in their *tensor* (Kronecker) product

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}, \qquad \text{e.g.} \qquad |0\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} =: |01\rangle. \tag{1.13}$$

Each new qubit doubles the length of the vector, which is the memory wall of Section 1.1. This course writes the first qubit (qubit 0) on the *left* of a ket, as textbooks do; Qiskit uses the opposite order, a difference treated carefully in Chapter 3.

Simulation 1.5 – One state, three tools

The state $(|0\rangle + \mathrm{i}|1\rangle)/\sqrt{2}$ in NumPy, in Qiskit (as a `Statevector`) and in QuTiP (as a `Qobj`).

```
import numpy as np
import qutip as qt
from qiskit.quantum_info import Statevector

amps = np.array([1, 1j]) / np.sqrt(2)  # (|0> + i|1>)/sqrt(2)

sv = Statevector(amps)                 # Qiskit
psi = qt.Qobj(amps.reshape(2, 1))      # QuTiP column vector

p_np = np.abs(amps) ** 2
p_qk = sv.probabilities()
p_qt = np.abs(psi.full().ravel()) ** 2
print("NumPy  probabilities:", np.round(p_np, 4))
print("Qiskit probabilities:", np.round(p_qk, 4))
print("QuTiP  probabilities:", np.round(p_qt, 4))
same = np.allclose(sv.data, psi.full().ravel())
print("same amplitudes:", same)
```

Output.

```
NumPy  probabilities: [0.5 0.5]
Qiskit probabilities: [0.5 0.5]
QuTiP  probabilities: [0.5 0.5]
same amplitudes: True
```

What to observe. The three tools hold the same two amplitudes and agree on $|\alpha|^2 = |\beta|^2 = 0.5$. For one qubit no ordering question arises; from Chapter 3 on, every multi-qubit Qiskit result in these notes is converted to textbook order.

1.10 Setting up the simulators

Every simulation in these notes runs locally, with open-source software, and needs no account or internet connection once installed. The versions below were used to produce the printed outputs (Windows 11, Python 3.12.7); newer versions usually work, but printed results are only guaranteed for these.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

1. Install Python 3.12 from https://www.python.org or a distribution such as Miniconda.
2. Create and activate a virtual environment, then install the pinned packages:

```
python -m venv eee6516
eee6516\Scripts\activate          (Windows)
source eee6516/bin/activate       (macOS, Linux)
pip install numpy==2.5.3 scipy==1.18.1 matplotlib==3.11.2
pip install qiskit==2.5.2 qiskit-aer==0.17.2 qutip==5.3.1
```

3. Run Simulation 1.5. If it prints the three lines of probabilities shown there, the environment is ready.

Two further simulators, Cirq and PennyLane, are introduced in Lecture 7 as alternative ways of writing the same circuits.

1.11 More linear algebra: trace, commutators, Gram–Schmidt

Trace. The trace of a square matrix is the sum of its diagonal entries, $\operatorname{Tr} A = \sum_i A_{ii}$. It satisfies $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$, and for a Hermitian matrix it equals the sum of the eigenvalues (in Simulation 1.3, $\operatorname{Tr} A = 3 = 0 + 3$). The trace returns in Chapter 7.

Commutators. The commutator of two matrices is $[A, B] = AB - BA$. For example,

$$XZ = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad ZX = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad [X, Z] = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix} \neq 0.$$

Two Hermitian matrices commute exactly when they have a common orthonormal eigenbasis. This is a statement about shared eigenvectors, not about the matrices being 'perpendicular'; every matrix commutes with itself.

> **Common mistake ('Commuting means orthogonal').** A zero commutator says that the order of applying A and B does not matter, and (for Hermitian matrices) that they share an eigenbasis. It says nothing about an inner product between them. Z and $\left[\begin{smallmatrix} 2 & 0 \\ 0 & 5 \end{smallmatrix}\right]$ commute, and both are diagonal in the same basis.

Gram–Schmidt. Two linearly independent kets can be turned into an orthonormal basis: normalise the first, subtract from the second its component along the first, and normalise the remainder. From $|v_1\rangle = (1, 1)^{\mathsf{T}}$ and $|v_2\rangle = (1, 0)^{\mathsf{T}}$: $|e_1\rangle = |+\rangle$, then $|v_2\rangle - \langle e_1 \,|\, v_2\rangle |e_1\rangle = (\tfrac{1}{2}, -\tfrac{1}{2})^{\mathsf{T}}$, which normalises to $|e_2\rangle = |-\rangle$.

Cauchy–Schwarz. For all kets, $|\langle \phi \,|\, \psi\rangle|^2 \leq \langle \phi \,|\, \phi\rangle \langle \psi \,|\, \psi\rangle$; for normalised kets, $|\langle \phi \,|\, \psi\rangle| \leq 1$, with equality only when the two kets differ by a phase factor.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

1.12 Vector spaces in general

The kets of this chapter are columns of numbers, which is all the course needs. Mathematics and physics texts define a vector space abstractly, by axioms, and treat matrices as representations of basis-free linear operators [4, §2.1]. In that setting the adjoint is defined by $\langle \phi \,|\, A\psi \rangle = \langle A^\dagger \phi \,|\, \psi \rangle$, and it equals the conjugate transpose only when the matrix is written in an orthonormal basis. The proof of the spectral theorem (existence of a full orthonormal eigenbasis) proceeds by induction on the dimension, splitting off one eigenvector at a time [4, Box 2.2]. Infinite-dimensional spaces, needed for the harmonic oscillator of Chapter 9, require more care with convergence but obey the same rules for the states used there.

A short history. Feynman proposed simulating physics with quantum machines in a keynote talk at MIT's Endicott House in May 1981, published in 1982 [2]. Deutsch defined a universal quantum computer in 1985 [1]. Shor presented polynomial-time quantum algorithms for factoring and discrete logarithms in November 1994 [8], and Grover a quadratically faster quantum search in 1996 [3]. The word *qubit* is generally credited to Schumacher's paper on quantum coding, published in 1995 [7].

1.13 Summary

Summary

- Describing n qubits classically needs 2^n complex amplitudes, $16 \cdot 2^n$ bytes in double precision; this wall motivates quantum computers and limits what we can simulate.
- Quantum computers are known to help with factoring, search and quantum simulation, not with computing in general, and today's devices are noisy (NISQ).
- A qubit is a two-entry complex vector with $|\alpha|^2 + |\beta|^2 = 1$; unlike probabilities, amplitudes can cancel.
- Kets are columns, bras are conjugate rows, $\langle \phi \,|\, \psi \rangle = \sum_i \phi_i^* \psi_i$, and coefficients in an orthonormal basis are inner products.
- Operators are matrices; projectors $|\psi\rangle\langle\psi|$ satisfy $P^2 = P$; an orthonormal basis gives $\sum_i |e_i\rangle\langle e_i| = I$.
- Hermitian matrices ($A^\dagger = A$) have real eigenvalues and orthonormal eigenvectors, $A = \sum \lambda_i |v_i\rangle\langle v_i|$; unitary matrices ($U^\dagger U = I$) preserve inner products and have eigenvalues of modulus 1.
- Functions of Hermitian matrices act on eigenvalues: $\mathrm{e}^{-\mathrm{i}\theta X/2} = \cos\frac{\theta}{2} I - \mathrm{i}\sin\frac{\theta}{2} X$.

Further reading: Nielsen and Chuang [4, §§1.1–1.2, 2.1] for the linear algebra in the same notation.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

1.14 Multiple-choice questions

Question 1.1. A simulator stores each amplitude in 16 B. The memory needed for the state vector of 30 qubits is

(a) 480 B
(b) 128 MiB
(c) 1 GiB
(d) 16 GiB

Question 1.2. For $z = 3 - 4\mathrm{i}$, the value of $|z|^2$ is

(a) 5
(b) -7
(c) 25
(d) $-7 - 24\mathrm{i}$

Question 1.3. The bra of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is

(a) $\begin{bmatrix}\alpha & \beta\end{bmatrix}$
(b) $\begin{bmatrix}\alpha^* \\ \beta^*\end{bmatrix}$
(c) $\begin{bmatrix}\alpha & \beta^*\end{bmatrix}$
(d) $\begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}$

Question 1.4. The real constant $c > 0$ that normalises $c\,(|0\rangle + 2\mathrm{i}|1\rangle)$ is

(a) $1/\sqrt{5}$
(b) $1/5$
(c) $1/\sqrt{3}$
(d) no real c exists, because $1^2 + (2\mathrm{i})^2 < 0$

Question 1.5. The inner product $\langle + \,|\, - \rangle$ equals

(a) 1
(b) 0
(c) $\frac{1}{2}$
(d) -1

Question 1.6. In $|0\rangle = c_+|+\rangle + c_-|-\rangle$, the coefficient c_+ is

(a) $\frac{1}{\sqrt{2}}$
(b) $\frac{1}{2}$
(c) 1
(d) 0

Question 1.7. Which matrix is unitary?

(a) $\begin{bmatrix}1 & 1 \\ 1 & 1\end{bmatrix}$
(b) $\begin{bmatrix}1 & 1 \\ 0 & 1\end{bmatrix}$

© 2026 S. M. Choudhury • Working draft • Not for redistribution

(c) $\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$

(d) $\begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{bmatrix}$

Question 1.8. Which statement is true?

(a) Every Hermitian matrix has real eigenvalues.
(b) Every square matrix has real eigenvalues.
(c) Every unitary matrix has real eigenvalues.
(d) Every Hermitian matrix has positive eigenvalues.

Question 1.9. For a normalised $|\psi\rangle$, the projector $P = |\psi\rangle\langle\psi|$ satisfies

(a) $P^2 = I$
(b) $P^2 = P$ and $P^\dagger = P$
(c) P has eigenvalues $+1$ and -1
(d) P is invertible

Question 1.10. Two Hermitian matrices satisfy $[A, B] = 0$. This means that

(a) A and B are orthogonal
(b) $AB = 0$
(c) A and B have a common orthonormal eigenbasis
(d) $A = B$

1.15 Problems

Problem 1.1 (Phasor arithmetic). Let $\alpha = \frac{1}{2}\mathrm{e}^{\mathrm{i}\pi/3}$ and $\beta = \frac{\sqrt{3}}{2}\mathrm{e}^{-\mathrm{i}\pi/4}$.

(a) Write α and β in Cartesian form.
(b) Show that $\alpha|0\rangle + \beta|1\rangle$ is normalised.
(c) Compute $\alpha^*\beta$ in polar form.

Problem 1.2 (Inner-product facts). [DERIVATION]

(a) From Equation (1.7), prove $\langle\psi\,|\,\phi\rangle = \langle\phi\,|\,\psi\rangle^*$.
(b) Show that $\{|\pm\mathrm{i}\rangle\} = \{(|0\rangle \pm \mathrm{i}|1\rangle)/\sqrt{2}\}$ is an orthonormal basis.
(c) Express $|1\rangle$ in that basis.

Problem 1.3 (Projectors in the $\pm$ basis).

(a) Write $|+\rangle\langle+|$ and $|-\rangle\langle-|$ as 2×2 matrices.
(b) Verify $P^2 = P = P^\dagger$ for both, and $|+\rangle\langle+| + |-\rangle\langle-| = I$.
(c) Compute $|+\rangle\langle+|\,|0\rangle$ and interpret it geometrically with Figure 1.4.

Problem 1.4 (Two properties of unitaries). [DERIVATION]

(a) Prove that the product of two unitary matrices is unitary.
(b) Prove that the columns of a unitary matrix form an orthonormal basis.
(c) Show that if A is Hermitian, $\mathrm{e}^{-\mathrm{i}A}$ is unitary.

© 2026 S. M. Choudhury • Working draft • Not for redistribution

Problem 1.5 (Spectral decomposition of Y). For $Y = \left[\begin{smallmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{smallmatrix}\right]$:

(a) find the eigenvalues and normalised eigenvectors;

(b) write Y in the form (1.12);

(c) compute $\mathrm{e}^{-\mathrm{i}\theta Y/2}$ and show that it is a real matrix.

Problem 1.6 (Norm preservation, numerically). [SIMULATION] Using NumPy with `np.random.default_rng(6516)`: build a random 2×2 unitary from the QR decomposition of a matrix with complex Gaussian entries, apply it to 1000 random normalised states, and report the largest deviation $|\,\|U\psi\| - 1\,|$. Then do the same with the non-unitary matrix N of Worked Example 1.5. Explain the size of each result.

Problem 1.7 (Three tools, one state). [SIMULATION] Extend Simulation 1.5 to the state of Worked Example 1.2. Create it in NumPy, Qiskit and QuTiP, print the amplitudes and probabilities from each, and confirm they agree to 10^{-12}. Then call `Statevector.draw('text')` and `print(psi)` for the QuTiP object and describe how each tool displays the state.

Problem 1.8 (A coupler as a quantum gate). [CHALLENGE ⋆] An ideal lossless 50:50 coupler (a 3 dB hybrid) can be modelled, with a suitable choice of reference planes, by $C = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & \mathrm{i} \\ \mathrm{i} & 1 \end{smallmatrix}\right]$.

(a) Show that C is unitary.

(b) Find diagonal phase matrices D_1 and D_2, with entries of modulus 1, such that $D_1 C D_2 = H$.

(c) Interpret the result: what does a phase shifter correspond to in matrix terms?

© 2026 S. M. Choudhury • Working draft • Not for redistribution

References

[1] D. Deutsch. ‘Quantum theory, the Church–Turing principle and the universal quantum computer’. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070.

[2] R. P. Feynman. ‘Simulating physics with computers’. In: *International Journal of Theoretical Physics* 21.6-7 (1982), pp. 467–488. DOI: 10.1007/BF02650179.

[3] L. K. Grover. ‘A fast quantum mechanical algorithm for database search’. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[4] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. 10th anniversary. Cambridge: Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[5] D. M. Pozar. *Microwave Engineering*. 4th ed. Hoboken, NJ: Wiley, 2012.

[6] J. Preskill. ‘Quantum computing in the NISQ era and beyond’. In: *Quantum* 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79.

[7] B. Schumacher. ‘Quantum coding’. In: *Physical Review A* 51.4 (1995), p. 2738. DOI: 10.1103/PhysRevA.51.2738.

[8] P. W. Shor. ‘Algorithms for quantum computation: discrete logarithms and factoring’. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

© 2026 S. M. Choudhury • Working draft • Not for redistribution