Emulating Quantum System with Libsolace

Paul Joo-Hyun Kim

January 17, 2026

Contents

1 Preface

This document is to introduce how the basics of quantum computing works from ground up “in a nutshell”, and how it is implemented in libsolace.

We will skip most of the discussion on quantum theory (such as the Schrödinger’s equation) and will only mention necessary concepts. I do highly recommend looking into those if you are interested/wanting to understand it better.

2 Mathematics of Quantum Computing

2.1 Prerequisites

To follow along this document, it would be very helpful to have understanding of the following concepts (which I will introduce shortly.)

- Linear Algebra
  - Vector, inner products, and norms.

  - The relationship between a matrix and a linear transformation.
  - Matrix multiplication and inversion
  - Orthogonal/Unitary Matrix
- Probability
  - Probability Distribution

2.1.1 Linear Algebra

We say an of complex numbers in $\mathbb{C}$ a **vector**. Here are some examples of vectors.

$$\begin{pmatrix} 1 \\ i \end{pmatrix} \in \mathbb{C}^2$$

$$\begin{pmatrix} 1 \\ \frac{1+i}{\sqrt{2}} \\ i \end{pmatrix} \in \mathbb{C}^3$$

Canonical vectors[1] is an important class of vectors where it has only one nonzero entry, and that nonzero entry is 1. For example,

$$e_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \in \mathbb{C}^n$$

$$e_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \in \mathbb{C}^n$$

The length n depends on the circumstance. Note that any non-canonical vector can be written as a linear combination[2] of canonical vectors as they form a basis. For example,

$$\begin{pmatrix} 3 \\ \frac{1+\sqrt{3}i}{2} \\ -i \end{pmatrix} = 3e_0 + \frac{1+\sqrt{3}i}{2}e_1 - ie_2 \in \mathbb{C}^3$$

Given two vectors u, v within $\mathbb{C}^n$, define the following operation.

$$\langle u|v\rangle = \sum_{i=0}^{n-1} \overline{u_i} v_i$$

[1] Often case denoted by lowercase alphabet

[2] Notice that I wrote I started indexing from 0. This is a computer science convention rather than mathematical convention. This allows for simplicity when working with modulo operations.

We will refer to this as **generalized inner product**, or simply inner product.

For a vector v, taking the inner product with itself results in the square of the "Euclidean length" of the vector:

$$|v|^2 = \langle v|v\rangle = \sum_{i=0}^{n-1} |v_i|^2$$

The **norm** $|v|$ of the vector v is precisely defined in terms of the generalized inner product. Note that a norm can only be nonnegative, and zero if and only if v has only zeros as its entries.

We also have an $m \times n$ (two-dimensional) array holding complex values. This is called a **matrix**[3]

$$\begin{pmatrix} 3 & 2 \\ 0 & i \end{pmatrix} \in \mathbb{C}^{2\times 2}$$

$$\begin{pmatrix} 2 & 1 & i \\ i & -3 & 0 \end{pmatrix} \in \mathbb{C}^{2\times 3}$$

$$\begin{pmatrix} 2 & 1 \\ i & i \\ -3 & 0 \end{pmatrix} \in \mathbb{C}^{3\times 2}$$

For quantum computing we are only interested in square matrices, that is an element of $\mathbb{C}^{n\times n}$ for some n.

From now on, we may assume all matrices we will be working on are square matrices unless stated othewise.

For matrices, we have **transpose** operator for "flipping" the entries across diagonal, turning a matrix of $m \times n$ into a matrix of $n \times m$.

$$\begin{pmatrix} 2 & 1 \\ i & i \\ -3 & 0 \end{pmatrix}^T = \begin{pmatrix} 2 & i & -3 \\ 1 & i & 0 \end{pmatrix}$$

More concretely, we can describe[4] it as

$$\left(A^T\right)_{i,j} = A_{j,i}$$

We also have **conjugate transpose**, which is to replace every element inside a matrix by complex conjugate, then taking the transpose.

$$\begin{pmatrix} 1+i & 1-i \\ 3+2i & e^{i\theta} \end{pmatrix}^* = \begin{pmatrix} 1-i & 3-2i \\ 1+i & e^{-i\theta} \end{pmatrix}$$

We define **matrix multiplication** between an array A of dimension $m \times n$ and B of dimension $n \times k$ as:

$$(AB)_{i,j} = \sum_{k=0}^{n-1} A_{i,k} B_{k,j}$$

[3] Often case denoted by uppercase alphabet.

[4] By convention, we write the index of the row first. We will start indexing from 0 throughout the document.

for $i = 0, \cdots, m-1$ and $j = 0, \cdots, k-1$. The resulting array AB is of shape $m \times k$. Note that $AB \neq BA$ in general (often times the operation itself is not defined for the other multiplication.)

It turns out multiplying a canonical vector e_j on the left by a square matrix A simply extracts the j^{th} column of A.

$$
\begin{aligned}
A &= \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \\
e_1 &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
Ae_1 &= \begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix}
\end{aligned}
$$

In general, if we can decompose a general vector v as linear combination of canonical vectors as the following,

$$v = \sum_{i=0}^{n-1} c_i e_i$$

then it turns out we can write the product of the matrix and vector as a linear combination analogously.

$$Av = \sum_{i=0}^{n-1} c_i \underbrace{Ae_i}_{\text{col } i \text{ of } A}$$

As you can see, we can think of matrix A as a function that maps a vector to another vector by a linear combination, that is, left multiplication by A can be thought of as a way to capture **linear transformation**.

Note that multiplying the **identity matrix** of the following form is like multiplying by one in numbers; it returns the other matrix.

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Given Av, it is also possible to find v back if A is known to be "invertible". This is known as **matrix inversion**. For example,

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

turns out to be invertible, meaning there exists some matrix A^{-1} such that it can "undo" the operation of multiplying by A. Explicitly, this is

$$A^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{2}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

For example, if given a general vector v, then $A^{-1}Av = v$.

Given an invertible matrix A, note that $A^{-1}A = I$ and also $AA^{-1} = I$.

If a real matrix Q satisfies the following, we say Q is an **orthogonal matrix**:

$$Q^T Q = QQ^T = I$$

Analogously, if a complex matrix Q satisfies the following, we say Q is a **unitary matrix**:

$$Q^* Q = QQ^* = I$$

Note that if a real matrix is an orthogonal matrix, it is obviously a unitary matrix.

Unitary matrices do not change the norm of a vector when multiplied, that is, for any general $v \in \mathbb{C}^n$ and unitary $Q \in \mathbb{C}^{n \times n}$,

$$|Qv| = |v|$$

This can easily be shown by noting the fact that,

$$\begin{aligned} |Qv|^2 &= (Qv)^* (Qv) \\ &= v^* Q^* Qv \\ &= v^* v \\ &= |v|^2 \end{aligned}$$

where the second line comes from the fact that applying transpose (or conjugate transpose) swaps the order of multiplication.

As an intuition, a unitary matrix encapsulates a linear transformation of which fixes the zero vector at zero vector and preserves lengths (and angles), such as rotation and flip.

Also, unitary matrices **ARE** the quantum gates, as it will be discussed later.

2.2 Probability

For quantum computing, the idea of probability distribution is important.

2.2.1 Probability Distribution

Suppose X is a random variable that can be any of the values within $\{0, 1, 2, \cdots, N\}$. We write $p(X = i)$ for the probability that if you were to observe random variable X, it is found to be i.

Since the law of total probability states that

$$\sum_{i=0}^{N} p(X = i) = 1$$

2.3 Quantum Computing

Finally, we get into quantum computing.

2.3.1 Qubit and State Vector

Just like classical computers where the smallest unit of information is composed of **bits**, that is, a container for 0 or 1, the smallest storage unit of information in a quantum computer is a **qubit**. A lot of resources may say that *a qubit is both 0 and 1 at the same time with some amount of probability*, but frankly, this is a misleading description; a qubit will always either be 0 or 1, not both. It is just that the result of observing a qubit is nondeterministic.

Just as we represent the OFF state of a bit with 0, and ON state of a bit with 1 in a classical computer, we represent the two states on a quantum computer with the following notation: $|0\rangle, |1\rangle$.

Mathematically speaking, you can think of them as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We now postulate that each qubit holds a thing called **state vector** $|\psi\rangle$ of the following form

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$$

where it satisfies the two conditions:

- $a_0, a_1 \in \mathbb{C}$
- $|a_0|^2 + |a_1|^2 = 1$

The second condition enforces the fact that a state vector ALWAYS has a norm of 1.

The heart of quantum computing is using "quantum gates" to manipulate state vector.

2.3.2 Measurement and Born Interpretation

A computer is not really useful if it does not give us a result back. In a classical computer, you can simply read the state of each bit as a result, but in a quantum computer, state vector is not a thing that we can see.

When we "measure" the value inside a qubit, it will always be either a $|0\rangle$ or a $|1\rangle$, but this will be a probablistic result.

The probability of measuring $|0\rangle$ and $|1\rangle$ from a qubit with statte vector $|\psi\rangle$ are described below,

$$\begin{aligned} P\left(|0\rangle\right) &= |\left\langle 0|\psi\right\rangle|^2 = |a_0|^2 \\ P\left(|1\rangle\right) &= |\left\langle 1|\psi\right\rangle|^2 = |a_1|^2 \end{aligned}$$

and a quirky thing about quantum computer is that, after we measure the qubit, the state vector undergoes **quantum state collapse**, where the entry of the state vector will change in a way that, when observed again, give the exact same result with 100% probability.

For example, suppose a qubit has the state vector $|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$. (NOTE: we cannot see the state vector in practice.) then the analysis above shows

$$\begin{aligned} P\left(|0\rangle\right) &= |\left\langle 0|\psi\right\rangle|^2 = \frac{1}{2} \\ P\left(|1\rangle\right) &= |\left\langle 1|\psi\right\rangle|^2 = \frac{1}{2} \end{aligned}$$

Suppose $|0\rangle$ was measured. Then the state vector collapses post-measurement to

$$|\psi\rangle = |0\rangle$$

that is, it no longer has probability to be measured as $|1\rangle$.

Extracting coefficient of the state vector and interpreting it as a component of probability, and the collapse after measurement is what is known as **Born interpretation**.

Some may be curious as to why this all happens. I would say, for that to be answered, you will need to consult real quantum physicsts.

For quantum information scientists, it is just how it works.

2.3.3 Quantum Gates

Suppose we are building a quantum circuit. It would not be of any use if the qubit keeps returning the same value over and over again (from quantum state collapse). This means, we need a way to "operate" on a qubit to make it do more interesting things.

To affect a qubit, we use what is known as a **quantum gate**. Mathematically, this is a unitary matrix $H \in \mathbb{C}^{2\times 2}$ that you can multiply to a state vector. Note that because left-multiplying by a unitary matrix does not change the norm of the state vector, the affected state vector is still a valid state vector.

Suppose we have a qubit that we recently observed $|0\rangle$ from. This means the qubit now has the state vector $|\psi\rangle$. Now consider the Hadamard gate, one of the basic quantum gates:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Applying this on the qubit, the state vector of the qubit changes,

$$H\left|\psi\right\rangle = H\left|0\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \left|1\right\rangle\right)$$

If we were to measure this qubit again, there is now an equal chance of measuring $\left|0\right\rangle$ and $\left|1\right\rangle$.

The goal of quantum computing now is to come up with quantum gates and to operate on qubits, in the way that it would allow us to get a meaning result with a sufficient probability.

3 Libsolace