Introduction to Quantum Computation
Summer 2020, University of Paderborn

Sevag Gharibian

Lecture 1: Introduction and Linear Algebra Review

I recall that during one walk Einstein suddenly stopped, turned to me and asked whether I really believed that the moon exists only when I look at it.
— Abraham Pais.

1 Introduction

Welcome to Introduction to Quantum Computation! In this course, we shall explore the subject of quantum computation from a theoretical computer science perspective. As the quote by Abraham Pais above foreshadows, our story will involve surprising twists and turns, which will seem completely at odds with your perception of the world around you. Indeed, in a quantum world, a single particle can be in two places simultaneously; two particles can be so strongly "bound" that they can *appear* to communicate instantaneously even if they are light-years apart; and the very act of "looking" at a quantum system can irreversibly alter the system itself! It is precisely these quirks of quantum mechanics which we shall aim to exploit in our study of computation.

The basic premise of quantum computing is "simple": To build a computer whose bits are not represented by transistors, but by subatomic particles such as electrons or photons. In this subatomic world, the pertinent laws of physics are no longer Newton's classical mechanics, but rather the laws of *quantum mechanics*. Hence, the name "quantum computing". Why would we ever want to build such a computer? There are a number of reasons. From an engineering standpoint, microchip components have become so small that they encounter quantum effects which impede their functionality. To a physicist, the study of quantum computing is a natural approach for simulating and studying quantum systems in nature. And to a computer scientist, quantum computers are remarkable in that they can solve problems which are believed to be intractable on a classical computer!

The field of quantum computing, which arguably started with famed physicist Richard Feynman's ideas (1982) for efficiently simulating physical systems (although it should be noted that ideas for crytography based on quantum mechanics date back to Stephen Wiesner around 1970), is nowadays far too large to be captured in a single course. Here, we shall focus on a broad introduction which aims to cover topics such as: What is quantum mechanics, and how can it be harnessed to build a computer? What kind of computational problems can such a computer solve? Are there problems which are hard even for a quantum computer? And finally, what does the study of quantum computing tell us about nature itself? Even if this course is the last time you encounter the topic of quantum computing, the experience should hopefully leave you with an appreciation for the fine interplay between the limits of physics and computing, as well as strengthen your background in Linear Algebra, which is useful in many other areas of computer science.

The entire course will take place in the mathematical framework of Linear Algebra, which we now review. It is crucial that you familiarize yourself with these concepts before proceeding with the course. These notes contain many exercises intended to help the reader; it is strongly recommended for you to work on these as you read along.

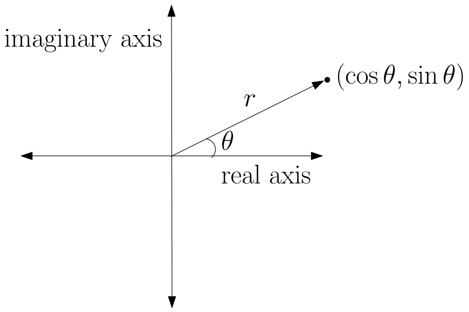
2 Linear Algebra

This course assumes a basic background in Linear Algebra. Thus, much of what is covered in this section is intended to be a refresher (although some of the later concepts here may be new to you); we thus cover this section briskly. Throughout this course, the symbols $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the sets of complex, real, integer, and natural numbers, respectively. For m a positive integer, the notation $[m]$ indicates the set $\{1, \ldots, m\}$.

The basic objects we shall work with are complex vectors $|\psi\rangle \in \mathbb{C}^d$, i.e.

$$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_d \end{pmatrix}, \tag{1}$$

for $\psi_i \in \mathbb{C}$. Recall here that a complex number $c \in \mathbb{C}$ can be written in two equivalent ways: Either as $c = a + bi$ for $a, b \in \mathbb{R}$ and $i^2 = -1$, or in its *polar form* as $c = re^{i\theta}$ for $r, \theta \in \mathbb{R}$. One of the advantages of the polar form is that it can directly be visualized on the 2D complex plane:



Here, the x and y axes correspond to *real* and *imaginary* axes, and r denotes the *length* of the vector $(\cos\theta, \sin\theta)$. For example, observe that 1 can be written in polar form with $r = 1$ and $\theta = 0$, i.e. 1 is represented in the 2D plane as vector $(1, 0)$. In this course, the polar form will be used repeatedly. Recall also that the complex conjugate of c, denoted c^*, is defined as $a - bi$ or $re^{-i\theta}$, respectively. Finally, the notation $|\cdot\rangle$ is called *Dirac* notation, named after physicist Paul Dirac, and is simply a useful convention for referring to column vectors. The term $|\psi\rangle$ is read "ket ψ".

Exercise. The *magnitude* or "length" of $c \in \mathbb{C}$ is given by $|c| = \sqrt{cc^*}$. What is the magnitude of $e^{i\theta}$ for any $\theta \in \mathbb{R}$? How about the magnitude of $re^{i\theta}$?

Complex vectors shall be crucial to us for a single reason: They represent quantum states (more details in subsequent lectures). It is thus important to establish some further basic properties of vectors. First, the *conjugate transpose* of $|\psi\rangle$ is given by

$$\langle\psi| = (\psi_1^*, \psi_2^*, \ldots, \psi_d^*), \tag{2}$$

where $\langle\psi|$ is a *row* vector. The term $\langle\psi|$ is pronounced "bra ψ". This allows us to define how much two vectors "overlap" via the *inner product* function, defined as

$$\langle\psi|\phi\rangle = \sum_{i=1}^{d} \psi_i^* \phi_i. \tag{3}$$

The inner product satisfies $(\langle\psi|\phi\rangle)^* = \langle\phi|\psi\rangle$. The "length" of a vector $|\psi\rangle$ can now be quantified by measuring the overlap of $|\psi\rangle$ with itself, which yields the *Euclidean norm*, $\parallel |\psi\rangle \parallel_2 = \sqrt{\langle\psi|\psi\rangle}$.

Exercise. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1, i)^T \in \mathbb{C}^2$, where T denotes the transpose. What is $\langle\psi|$? How about $\parallel |\psi\rangle \parallel_2$?

With a norm in hand, we can define a notion of *distance* between vectors $|\psi\rangle, |\phi\rangle$, called the Euclidean distance: $\parallel |\psi\rangle - |\phi\rangle \parallel_2$. This distance will play the important role of quantifying how well two quantum states $|\psi\rangle$ and $|\phi\rangle$ can be "distinguished" via measurements. Two useful properties of the Euclidean norm are:

1. (Positive scalability) $\parallel a|\psi\rangle \parallel_2 = |a| \parallel |\psi\rangle \parallel_2$ for $a \in \mathbb{C}$.

2. (Triangle inequality) For any $|\psi\rangle, |\phi\rangle$, one has $\| \, |\psi\rangle + |\phi\rangle \, \|_2 \leq \| \, |\psi\rangle \, \|_2 + \| \, |\phi\rangle \, \|_2$.

These two properties can be used to show that for all $|\psi\rangle \in \mathbb{C}^d$, $\| \, |\psi\rangle \, \|_2 \geq 0$.

Exercise. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1, i)^T \in \mathbb{C}^2$ and $|\phi\rangle = (\frac{1}{2}, \frac{\sqrt{3}}{2})^T \in \mathbb{C}^2$, where T denotes the transpose. What is $\| \, |\psi\rangle - |\phi\rangle \, \|_2$?

Orthonormal bases. A much more natural way to represent vectors in this course shall be in terms of *orthonormal bases*. Recall that a set of vectors $\{|\psi\rangle_i\} \subseteq \mathbb{C}^d$ is *orthogonal* if for all $i \neq j$, $\langle\psi|_i|\psi\rangle_j = 0$, and *orthonormal* if $\langle\psi|_i|\psi\rangle_j = \delta_{ij}$. Here, δ_{ij} is the Kroenecker delta, whose value is 1 if $i = j$ and 0 otherwise. For the vector space $\mathbb{C}^d$, which has dimension d, it is necessary and sufficient to use d orthonormal vectors in order to form an orthonormal basis.

One of the most common bases we use is the *computational basis*, defined for $\mathbb{C}^2$ as

$$|0\rangle = \left(\begin{array}{c} 1 \\ 0 \end{array} \right) \qquad\qquad |1\rangle = \left(\begin{array}{c} 0 \\ 1 \end{array} \right). \tag{4}$$

Since $\{|0\rangle, |1\rangle\}$ is an orthonormal basis, any vector $|\psi\rangle \in \mathbb{C}^2$ can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ for some $\alpha, \beta \in \mathbb{C}$. We say $|\psi\rangle$ is *normalized* when it has length 1, i.e. $\| \, |\psi\rangle \, \|_2 = 1$; equivalently, this means $|\alpha|^2 + |\beta|^2 = 1$.

Exercise. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1, 1)^T \in \mathbb{C}^2$. Write $|\psi\rangle$ in terms of the computational basis for $\mathbb{C}^2$. Is $|\psi\rangle$ normalized?

The computational basis is easily extended to d-dimensional vectors by defining $|i\rangle \in \mathbb{C}^d$ as having a 1 in position i and 0 elsewhere (here, $0 \leq i \leq d-1$). In this course, vectors labelled by integers (e.g. $|1\rangle$, $|3\rangle \in \mathbb{C}^d$) will be assumed to be d-dimensional computational basis vectors.

Linear maps. Given a vector $|\psi\rangle \in \mathbb{C}^d$, we are interested in how $|\psi\rangle$ can be "mapped" to other vectors. The maps we consider are *linear*, which by definition means that for map $\Phi : \mathbb{C}^d \mapsto \mathbb{C}^d$ and arbitrary $\sum_i \alpha_i|\psi_i\rangle \in \mathbb{C}^d$,

$$\Phi\left(\sum_i \alpha_i|\psi_i\rangle\right) = \sum_i \alpha_i\Phi(|\psi_i\rangle). \tag{5}$$

The set of linear maps from vector space $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathcal{L}(\mathcal{X}, \mathcal{Y})$. For brevity, we use the shorthand $\mathcal{L}(\mathcal{X})$ to mean $\mathcal{L}(\mathcal{X}, \mathcal{X})$.

Exercise. Consider the linear map $\Phi : \mathbb{C}^2 \mapsto \mathbb{C}^2$ with action $\Phi(|0\rangle) = |1\rangle$ and $\Phi(|1\rangle) = |0\rangle$. If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, what is $\Phi(|\psi\rangle)$?

The exercise above teaches us an important lesson — the action of a linear map $\Phi \in \mathcal{L}(\mathbb{C}^d)$ is fully characterized by understanding how Φ acts on a basis for $\mathbb{C}^d$. This leads to a natural representation for Φ in terms of a *matrix*.

Recall that a $d \times d$ *matrix* A is a two-dimensional array of complex numbers whose (i, j)th entry is denoted $A(i, j) \in \mathbb{C}$ for $i, j \in [d]$. To represent a linear map $\Phi : \mathbb{C}^d \mapsto \mathbb{C}^d$ as an $d \times d$ matrix A_Φ, we use its action on a basis for $\mathbb{C}^d$. Specifically, define the ith column of A_Φ as $\Phi(|i\rangle)$ for $\{|i\rangle\}$ the standard basis for $\mathbb{C}^d$, or

$$A_\Phi = \left[\begin{array}{c} \Phi(|0\rangle), \Phi(|1\rangle), \ldots, \Phi(|d-1\rangle) \end{array} \right]. \tag{6}$$

In this course, we use both the matrix and linear map views interchangeably, with the application notion clear from context.

Exercise. What is the 2×2 complex matrix representing the linear map Φ from the previous exercise? What is the linear map whose matrix (with respect to the computational basis) is the *identity matrix*

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}? \tag{7}$$

The product AB of two $d \times d$ matrices A and B is also a $d \times d$ matrix with entries

$$AB(i,j) = \sum_{k=1}^{d} A(i,k)B(k,j). \tag{8}$$

Note that unlike for scalars, for matrices it is *not* always true that $AB = BA$. In the special case where $AB = BA$, we say A and B *commute*.

Exercise. Define

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{9}$$

Do X and Z commute?

We would like to understand how "large" the output space of a linear map $A \in \mathcal{L}(\mathbb{C}^d)$ is. To this end, the *image* of A is the set of all possible output vectors under the action of A, i.e.

$$\mathrm{Im}(A) := \left\{ |\psi\rangle \in \mathbb{C}^d \mid |\psi\rangle = A|\phi\rangle \text{ for some } |\phi\rangle \in \mathbb{C}^d \right\}. \tag{10}$$

The *rank* of A is the dimension of its image.

Exercise. Suppose $\mathrm{rank}(A) = 0$. What is $\mathrm{Im}(A)$? How about the case of $\mathrm{rank}(A) = d$?

The set of all vectors sent to zero by A is called its *null space*, i.e. $\mathrm{Null}(A) := \left\{ |\psi\rangle \in \mathbb{C}^d \mid A|\psi\rangle = 0 \right\}$. It holds that $\dim(\mathrm{Null}(A)) + \dim(\mathrm{Im}(A)) = d$ (here dim denotes dimension).

Exercise. Is there a non-zero vector in the null space of matrix Z from Equation (9)? (Hint: Multiply an arbitrary vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ by Z and see if you can make the zero vector pop out.) What does the answer tell you about $\mathrm{rank}(Z)$? What is the null space of matrix

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{11}$$

and what is $\mathrm{rank}(B)$?

Matrix operations. Matrices encode the operations which we are allowed to perform on our vectors. There are some simple operations on matrices *themselves* which will show up repeatedly in our discussions. The first three of these are the linear maps *complex conjugate*, *transpose* and *adjoint*, defined respectively as

$$A^*(i,j) = (A(i,j))^* \qquad A^T(i,j) = A(j,i) \qquad A^\dagger = (A^*)^T. \tag{12}$$

Note that $(AB)^\dagger = B^\dagger A^\dagger$, and similarly for the transpose. These operations apply to vectors as well so that $\langle\psi|$, defined in Equation (2), is simply $|\psi\rangle^\dagger$. The adjoint will especially play a crucial role in this course.

Exercise. Calculate $X^\dagger$ and $Z^\dagger$ for X and Z from Equation (9), as well as the adjoint of

$$A = \begin{pmatrix} 1 & 2 \\ 3 & e^{i\pi/2} \end{pmatrix}. \tag{13}$$

Another useful function on matrices is the *trace*, which is simply a linear map $\mathrm{Tr} : \mathcal{L}(\mathbb{C}^d) \mapsto \mathbb{C}$ summing the entries on the diagonal of A, i.e. $\mathrm{Tr}(A) = \sum_{i=1}^d A(i,i)$. A wonderful property of the trace is that it is *cyclic*, i.e. $\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB)$. This implies that even if A and B do not commute, i.e. $AB \neq BA$, it nevertheless holds that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$!

Exercise. In a previous exercise, you showed that X and Z do not commute. Compute $\mathrm{Tr}(XZ)$ and $\mathrm{Tr}(ZX)$ and verify that they are indeed the same.

Outer products. The Dirac notation lends itself particularly well to an alternate description of matrices via *outer products*. For vectors $|\psi\rangle, |\phi\rangle \in \mathbb{C}^d$, the outer product is $|\psi\rangle\langle\phi| \in \mathcal{L}(\mathbb{C}^d)$; unlike the inner product, the outer product yields a $d \times d$ matrix. It can be computed straightforwardly via the rules of matrix multiplication. For example,

$$|0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad |1\rangle\langle 0| = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{14}$$

More generally, the matrix $|i\rangle\langle j| \in \mathcal{L}(\mathbb{C}^d)$ has a 1 at position (i,j) and zeroes elsewhere. This yields a simple yet neat trick: A matrix $A \in \mathcal{L}(\mathbb{C}^d)$ written in the computational basis can hence be expressed as $\sum_{ij} A(i,j)|i\rangle\langle j|$. It is thus easy to evaluate expressions of the form

$$\langle i|A|j\rangle = \langle i| \left(\sum_{i'j'} A(i',j')|i'\rangle\langle j'| \right) |j\rangle = \sum_{i'j'} A(i',j')\langle i|i'\rangle\langle j|j'\rangle = \sum_{i'j'} A(i',j')\delta_{ii'}\delta_{jj'} = A(i,j), \tag{15}$$

where the third equality follows since $\{|i\rangle\}$ forms an orthonormal basis for $\mathbb{C}^d$. In other words, $\langle i|A|j\rangle$ simply rips out entry $A(i,j)$! These types of expressions will be ubiquitous in the setting of quantum measurements.

Exercise. Observe that X from Equation 9 can be written $X = |0\rangle\langle 1| + |1\rangle\langle 0|$. What is $\langle 0|X|0\rangle$? How about $\langle 0|X|1\rangle$? How can you rewrite $\mathrm{Tr}(X)$ in terms of expressions of this form?

Eigenvalues and eigenvectors. With outer products in hand, we can discuss one of the most fundamental tools in our Linear Algebraic toolkit — eigenvalues and eigenvectors. Given any matrix $A \in \mathcal{L}(\mathbb{C}^d)$, an *eigenvector* is a special non-zero vector satisfying the equation

$$A|\psi\rangle = \lambda|\psi\rangle, \tag{16}$$

for some $\lambda \in \mathbb{C}$ which is the corresponding *eigenvalue*.

Exercise. Show that $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ are both eigenvectors of X from Equation (9). What are their respective eigenvalues?

The outer product can now be used to state an expression which will be used repeatedly in this course. For any matrix A satisfying $AA^\dagger = A^\dagger A$ (such matrices are called *normal*; most matrices in this course will be normal), we can decompose A in terms of its eigenvalues and eigenvectors as

$$A = \sum_{i=1}^d \lambda_i |\lambda_i\rangle\langle\lambda_i|, \tag{17}$$

where λ_i and $|\lambda_i\rangle$ are the eigenvalues and corresponding eigenvectors of A. This is called the *spectral decomposition* of A. The spectral decomposition is useful for a few reasons. First, it tells us exactly how A acts on $\mathbb{C}^d$; this is because the eigenvectors $|\lambda_i\rangle \in \mathbb{C}^d$ can be chosen[1] to form an orthonormal basis for $\mathbb{C}^d$. Thus, any vector $|\psi\rangle \in \mathbb{C}^d$ can be written in terms of the eigenvectors of A, i.e. $|\psi\rangle = \sum_i \alpha_i|\lambda_i\rangle$ for some $\alpha_i \in \mathbb{C}$. The spectral decomposition also immediately reveals the rank of A; specifically, rank(A) is just the number of non-zero eigenvalues of A. Finally, $\mathrm{Tr}(A)$ has a very simple expression in terms of A's eigenvalues — $\mathrm{Tr}(A) = \sum_i \lambda_i$. Let us quickly prove this last claim:

$$\mathrm{Tr}(A) = \mathrm{Tr}\left(\sum_i \lambda_i|\lambda_i\rangle\langle\lambda_i|\right) = \sum_i \lambda_i \mathrm{Tr}(|\lambda_i\rangle\langle\lambda_i|) = \sum_i \lambda_i \mathrm{Tr}(\langle\lambda_i|\lambda_i\rangle) = \sum_i \lambda_i. \tag{18}$$

Here, the second equality follows since the trace is linear, the third by the cyclic property of the trace, and the last since the eigenvectors $|\lambda_i\rangle$ are orthonormal.

Exercise. In the previous exercise, you computed the eigenvectors and eigenvalues of X. Use these to write down the spectral decomposition of X, and verify that it indeed evaluates to X. Next, recall that $X|0\rangle = |1\rangle$. Note that $|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$. Use this and the spectral decomposition of X to verify that indeed $X|0\rangle = |1\rangle$.

Finally, recall that the eigenvalues of $A \in \mathcal{L}(\mathbb{C}^d)$ can be computed as the roots of the degree-d *characteristic polynomial* of A, p_A, defined

$$p_A(\lambda) = \det(\lambda I - A), \tag{19}$$

where the determinant det can be defined recursively as

$$\det(A) = \sum_{j=1}^{d} (-1)^{i+j} A(i,j) \det(A_{ij}). \tag{20}$$

Here, A_{ij} is the matrix obtained from A by deleting row i and column j, and we define the base case of this recursion (i.e. a 1×1 matrix $[c]$) as $\det([c]) = c$. This equation holds for any $i \in [d]$. In the special case when $d = 2$, this reduces nicely to

$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc. \tag{21}$$

Exercise. Use Equations (19) and (21) to compute the eigenvalues of Z from Equation (9). Then, plug these back into Equation (16) to solve for the eigenvectors of Z. What is the spectral decomposition of Z?

Let us close with a simple observation: For any diagonal matrix A (written in the computational basis), the eigenvalues of A are simply the entries on the diagonal of A, and the eigenvectors are just the computational basis vectors. In the exercise above, this immediately confirms that the eigenvalues of Z are 1 and -1 with eigenvectors $|0\rangle$ and $|1\rangle$, respectively.

[1] This statement need not hold for non-normal matrices. In fact, one can prove that a matrix is normal if and only if it admits a spectral decomposition. Non-normal matrices do, however, admit the more general *singular value decomposition*.