Quantum Algorithms through Linear Algebra

Lecture Notes for CSE 410

Matthew G. Knepley

Department of Computer Science and Engineering
University At Buffalo

August 21, 2020

I dedicate these notes to my wonderful wife Margarete, without whose patience and help they could never have been written.

Acknowledgements

TBD

Contents

Chapter 1

Introduction

In this course, we will be mainly concerned with discrete binary systems, meaning those for which a measurement returns a 0 or 1, or true/false, heads/tails, up/down, Hall/Oates, or any other dichotomy you wish to use. It is definitely possible to use ternary systems, or k-ary, but almost all physical models of quantum computing use binary. We will write the state of such a system in at least two ways. First, we have the familiar linear-algebraic notation for a two-state system,

$$\text{down} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \text{up} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where each possible state is assigned a basis vector. Another popular notation, known as *Dirac notation*, names the basis vectors explicitly

$$\text{down} = |0\rangle \qquad \text{up} = |1\rangle\,,$$

or even

$$\text{down} = |d\rangle \qquad \text{up} = |u\rangle\,.$$

We could make our linear algebra look more like Dirac notation by using basis vectors $\hat{\mathbf{e}}_i$ explicitly

$$\text{down} = \hat{\mathbf{e}}_0 \qquad \text{up} = \hat{\mathbf{e}}_1.$$

We will call our two-state system a *bit*, which is a portmanteau of "binary digit". Claude E. Shannon first used the word bit in his seminal 1948 paper, A Mathematical Theory of Communication (Shannon 1948), and attributed its origin to John W. Tukey. Many times it will be helpful to think of an actual system, such as a coin. If the coin shows tails, we will say that it is in state $\hat{\mathbf{e}}_0$ or $|T\rangle$, but for heads it is in state $\hat{\mathbf{e}}_1$ or $|H\rangle$. For a normal, deterministic coin in order to specify the state, we merely give the appropriate basis vector, heads or tails. However, this is not the right setting for understanding the quantum analogue.

Instead, let us imagine flipping the coin. Now we have introduced indeterminacy, or stochasticity, into our system. What can we say about the state of our coin after flipping, before we lift our hand? We would probably say that it is equally likely to be heads or tails. In our notation above, we could write

$$\begin{aligned}\frac{1}{2}\text{down} + \frac{1}{2}\text{up} &= \frac{1}{2}|T\rangle + \frac{1}{2}|H\rangle \\ &= \frac{1}{2}\hat{\mathbf{e}}_0 + \frac{1}{2}\hat{\mathbf{e}}_1\end{aligned}$$

where we interpret the coefficient in front of each basis vector as "the chance our system ends up in this state". This can be thought of now as a probabilistic bit, or *probit*. The chance of observing a certain outcome $|j\rangle$ from state $|\psi\rangle$ is then $\langle j|\psi\rangle$, or using linear algebra $\hat{\mathbf{e}}_j^\dagger \boldsymbol{\psi}$. This is very close to the result for a quantum mechanical system, for which the chance of observation is the square of this quantity. We will see in later chapters that the proper classical analogues to quantum mechanical systems are probabilistic, not deterministic, classical systems.

Now suppose we want to change the probit system by changing the probabilities of getting either state, from $(\frac{1}{2}\,\frac{1}{2})$ to $(\,p_1\;p_2\,)$. Since the two states are the only possible outcomes, we would still want the sum of the two coefficients to be one, meaning that we are guaranteed to get one of them. If, in addition, we demand that the change be linear. we would have a matrix equation

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

where the matrix U has nonnegative entries, and its rows must sum to one, which is called *right stochastic matrix*. We note that this matrix preserves the 1-norm of the input vector $\vec{\mathbf{p}}$. We will see that quantum evolution uses unitary matrices that preserve the 2-norm of the input vectors of quantum amplitudes.

We can create a more abstract setting for the ideas we have discussed above. Let us call a *state* a description of our system which is sufficient to predict the outcome of any measurement, and the number of real parameters (or measurements) necessary to define the state will be the number of *degrees of freedom* K. The *dimension* of our system will be the maximum number of states which can be distinguished by a single measurement. For example, our single coin system has $K = 2$ degrees of freedom, p_0 and p_1, as well as dimension two, $p\,|0\rangle$ and $(1-p)\,|1\rangle$. Note that the dimension is also the number of basis vectors, matching the usual definition from linear algebra. We will call the *probability* of an event, the relative frequency of its occurrence when the measurement is performed on a ensemble of n identically prepared systems in the limit as n becomes infinite.

The important thing to explore is how composite systems behave, namely those that are composed of collections of simpler systems. Suppose that we have two coins. Then the possible configurations for this system are

$$|TT\rangle, |TH\rangle, |HT\rangle, |HH\rangle$$

so that it has dimension four. We will be able to specify the outcomes of measurements using four probability weights $(\,p_1\ p_2\ p_3\ p_4\,)$, so that $K = N$. We expect that if we fix one of the coins, then this system will behave just like a one probit system with a single coin, which is indeed that case. When we combined the two coins, we saw that $N_2 = N_1 \cdot N_1$ and likewise $K_2 = K_1 \cdot K_1$. We will take this as a general axiom for any two systems. Given these assumptions about the behavior of composite systems, one can prove (Hardy 2001; Schack 2003) that

$$K = N^r$$

where r is a positive integer. If one additional insists that a reversible, continuous transformation exist between the pure states of the system, we can rule out $r = 1$ since there are a finite number of pure states in this case. This situation is exactly classical probability theory, and the pure states correspond to the basis vectors. In quantum theory, we have an infinity of of pure states so that a continuous transformation between them is possible, and a full state is described by a *density matrix* which has N^2 real parameters.

1.1 The Quantum Mechanical Setting

Complex Hilbert space is the setting for quantum mechanics. Hardy (Hardy 2001) shows that this is related to the behavior of composite systems. In real Hilbert space, composite systems have too many degrees of freedom, and in quaternionic Hilbert space they have too few. The signature operation in a Hilbert space, the inner product $\langle\phi|psi\rangle$, is defined by three properties:

1. **Positivity**: $\langle\psi|\psi\rangle > 0 \qquad \forall\psi \neq 0.$
2. **Conjugate Symmetry**: $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle$
3. **Linearity**: $\langle\phi|\left(a\,|\psi\rangle + b\,|\omega\rangle\right) = a\,\langle\phi|\psi\rangle + b\,\langle\phi|\omega\rangle$

The complex Hilbert space $\mathcal{H}$ is complete in the norm induced by the inner product

$$\|\psi\|^2 = \langle\psi|\psi\rangle \tag{1.1}$$

With this space, we can now define the mathematical objects representing our physical objects. First we will state the common definitions. These are quite clean, but misleading, since they only represent isolated systems. We will see afterwards that for composite systems a more complicated approach is necessary.

States A *state* of our physical system, meaning the information sufficient to determine the probabilistic outcome of any measurement. This means that knowing the state and being able to prepare many identical systems in this state, I can predict the probability (long run average) of any measurement. Our state will be a *ray* in $\mathcal{H}$, meaning the equivalence class of vectors $a\psi$ or any scaling $a \in \mathbb{C}$, including scalings $e^{i\alpha}$ which preserve the norm, called a *phase*.

Observables An *observable* is a property of a physical system that can be measured. In quantum mechanics, an observable is a self-adjoint linear operator. The eigenstates of a self-adjoint linear operator for an orthonormal basis, and therefore A has a *spectral representation* in terms of these states ϕ_i

$$A = \sum_i \lambda_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right| = \sum_i \lambda_i P_i, \tag{1.2}$$

where P_i is the orthogonal projector onto the ith eigenspace.

Measurement When we measure A for a quantum state ψ, this collapses the system into an eigenstate ϕ of A and the value of the measurement is the eigenvalue λ. We will define a defineTermmeasurement of an eigenstate ϕ of observable A on the state ψ to be a map M from $A\left|\psi\right\rangle$ to the real numbers

$$M : \mathcal{H} \to \mathbb{R}$$

which gives the probability of obtaining state ϕ and value λ after the operation. This probability is given by

$$\Pr(\lambda) = \left\|P\left|\psi\right\rangle\right\|^2 = \left\langle\psi|P|\psi\right\rangle \tag{1.3}$$

Now we consider a composite system AB composed of two subsystems A and B. The state ψ_{AB} of this composite system lives in the Hilbert space $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$. For the next example, we will consider a two qubit state, but we can easily generalize this to bigger systems. If we want to measure observable L only the A qubit, our combined observable L will be

$$L = L_A \otimes I$$

where I is the identity operator on $\mathcal{H}_B$. If we look at the expectation value of L,

$$\begin{aligned}
\left\langle\psi|L|\psi\right\rangle &= \left(\bar{a}\left\langle 0\right| \otimes \left\langle 0\right| + \bar{b}\left\langle 0\right| \otimes \left\langle 1\right| + \bar{c}\left\langle 1\right| \otimes \left\langle 0\right| + \bar{d}\left\langle 1\right| \otimes \left\langle 1\right|\right)(L_A \otimes I) \\
&\quad \left(a\left|0\right\rangle \otimes \left|0\right\rangle + b\left|0\right\rangle \otimes \left|1\right\rangle + c\left|1\right\rangle \otimes \left|0\right\rangle + d\left|1\right\rangle \otimes \left|1\right\rangle\right) && (1.4) \\
&= |a|^2\left\langle 0|L_A|0\right\rangle + \bar{a}c\left\langle 0|L_A|1\right\rangle + |b|^2\left\langle 0|L_A|0\right\rangle + \bar{b}d\left\langle 0|L_A|1\right\rangle \\
&\quad + |c|^2\left\langle 1|L_A|1\right\rangle + \bar{c}a\left\langle 1|L_A|0\right\rangle + |d|^2\left\langle 1|L_A|1\right\rangle + \bar{d}b\left\langle 1|L_A|0\right\rangle && (1.5) \\
&= \left(|a|^2 + |b|^2\right)\left\langle 0|L_A|0\right\rangle + 2\,\mathrm{Re}\{\bar{a}c + \bar{b}d\}\left\langle 0|L_A|1\right\rangle \\
&\quad + \left(|c|^2 + |d|^2\right)\left\langle 1|L_A|1\right\rangle && (1.6)
\end{aligned}$$

This expression can be rewritten into a matrix equation

$$\begin{aligned}
\left\langle\psi|L|\psi\right\rangle &= \left(|a|^2 + |b|^2\right)\mathrm{Tr}\{L_A\left|0\right\rangle\!\left\langle 0\right|\} + \left(|c|^2 + |d|^2\right)\mathrm{Tr}\{L_A\left|1\right\rangle\!\left\langle 1\right|\} \\
&\quad + \mathrm{Re}\{\bar{a}c + \bar{b}d\}\,\mathrm{Tr}\{L_A\left|0\right\rangle\!\left\langle 1\right|\} + \mathrm{Re}\{\bar{a}c + \bar{b}d\}\,\mathrm{Tr}\{L_A\left|1\right\rangle\!\left\langle 0\right|\} && (1.7) \\
&= \mathrm{Tr}\left\{L_A\begin{pmatrix} |a|^2 + |b|^2 & \mathrm{Re}\{\bar{a}c + \bar{b}d\} \\ \mathrm{Re}\{\bar{a}c + \bar{b}d\} & |c|^2 + |d|^2 \end{pmatrix}\right\} && (1.8) \\
&= \mathrm{Tr}\{L_A\rho_A\} && (1.9)
\end{aligned}$$

where ρ_A is called the *density operator* for system A. Clearly the density operator is self-adjoint, and has real entries. It also has trace 1 since the initial combined state was normalized, and it has only positive eigenvalues (which might not be immediately clear).

Since this form for the expectation value of L is true for any observable acting on system A, we can interpret ρ_A as defining a statistical ensemble of quantum states for system A rather than a set of states linked to the states of system B. For example, suppose we have the simple state

$$a\,|0\rangle \otimes |0\rangle + b\,|1\rangle \otimes |1\rangle \tag{1.10}$$

which we call the *Bell state*, where the off-diagonal terms above vanish. Then the density operator is diagonal, with entries $|a|^2$ and $|b|^2$. The result of our expectation value is exactly what we would expect to get if we specified that system A was in state $|0\rangle$ with probability $p_0 = |a|^2$ and state $|1\rangle$ with probability $p_1 = |b|^2$. This is quite different from system A being in a superposition of states $|0\rangle$ and $|1\rangle$, as we illustrate with a small example.

Suppose we prepare a system in the Bell state from above with $|a|^2 = |b|^2 = \frac{1}{2}$, so that the density operator ρ_A is given by

$$\rho_A = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}I,$$

and it looks like an ensemble over the two equally probable states. We distinguish this from the single system in the superposition of states,

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

Then if we measure the probability for the state ψ_A from the ensemble state, we get

$$\begin{aligned}
\langle P_\psi \rangle &= \mathrm{Tr}\{|\psi_A\rangle\langle\psi_A|\,\rho_A\} && (1.11)\\
&= \frac{1}{4}\,\mathrm{Tr}\{(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|)\,(|0\rangle\langle 0| + |1\rangle\langle 1|)\} && (1.12)\\
&= \frac{1}{4}\,(1+1) && (1.13)\\
&= \frac{1}{2}, && (1.14)
\end{aligned}$$

whereas if we measured the probability of the original state ψ_A to be in state ψ_A, we would of course get unity, since it is certainly in that state. In fact, we may take any unitary transformation U of ψ_A and get the same result

$$\begin{aligned}
\langle P_{U\psi} \rangle &= \mathrm{Tr}\big\{|U\psi_A\rangle\langle\psi_A|\,U^\dagger\rho_A\big\} && (1.15)\\
&= \mathrm{Tr}\left\{|\psi_A\rangle\langle\psi_A|\,U^\dagger\frac{1}{2}IU\right\} && (1.16)\\
&= \mathrm{Tr}\{|\psi_A\rangle\langle\psi_A|\,\rho_A\}. && (1.17)
\end{aligned}$$

We can now define clearly a *pure state*, which is a single ray in the Hilbert space, or something with a density operator which has a single term (diagonal element) in the eigenbasis. We can also call a state with multiple terms in the diagonalized density matrix an *incoherent superposition*, as opposed to a *coherent superposition* which is the normal pure state we have seen before.

References

Shannon, Claude Elwood (July 1948). "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://web.archive.org/web/19980715013250/http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

Hardy, Lucien (2001). "Quantum theory from five reasonable axioms". In: eprint: quant-ph/0101012.

Schack, Rüdiger (2003). "Quantum theory from four of Hardy's axioms". In: *Foundations of Physics* 33.10, pp. 1461–1468.

Chapter 2

Linear Spaces

Numerical linear algebra is one of the most developed parts of numerical analysis. It is also the solid foundation of numerical computing. The basic outline of linear algebra has been clear since at least Grassman's 1862 treatment (Fearnley-Sander 1979). A vector space V over a field F is defined by the axioms in Table 2, and in everything we do F will be either the real or complex numbers. In addition, linear algebra studies mappings between vector spaces that preserve the vector-space structure. Given two vector spaces V and W, a linear operator is a map $A : V \to W$ that is compatible with vector addition and scalar multiplication,

$$A(\mathbf{u} + \mathbf{v}) = A\mathbf{u} + A\mathbf{v}, \quad A(a\mathbf{v}) = aA\mathbf{v} \qquad \forall \mathbf{u}, \mathbf{v} \in V, a \in F. \tag{2.1}$$

This should have been covered in detail in your linear algebra courses.

There are two principal jobs in scientific computing: design of the interface in order to control complexity, and efficiency of the implementation. In this unit we will try to indicate why the current interface has become the standard, and what pieces of it are likely to continue going forward. In a later unit, we will analyze the runtime performance of various implementations. However, none of this can be accomplished without the ability to run a linear algebra code.

There are many well-known packages which support numerical linear algebra, including BLAS/LAPACK (Lawson et al. 1979; Anderson et al. 1990), Hypre (Falgout 2017; Falgout n.d.), Trilinos (Heroux and Willenbring 2003; Heroux et al. n.d.), DUNE (Bastian et al. 2015), Eigen (Jacob and Guennebaud 2015), and Elemental (Poulson et al. 2013; Poulson 2015). We will use the PETSc libraries (Balay, Abhyankar, et al. 2020; Balay, Abhyankar, et al. 2019; Balay, Gropp, et al. 1997) for a number of reasons. PETSc supports scalable, distributed sparse linear algebra, which will be our focus since we will be concerned with larger problems that cannot be contained in a single machine memory and mainly with PDE or graph problems which have a sparse structure. For dense linear algebra problems, we will use Elemental. PETSc is designed as a hierarchical set of library interfaces, and uses C to enhance both portability

Axiom	Signification
Associativity of addition	$\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
Commutativity of addition	$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
Vector identity element	$\exists 0 \in V \mid \mathbf{v} + 0 = \mathbf{v} \quad \forall \mathbf{v} \in V$
Vector inverse element	$\forall \mathbf{v} \in V, \exists - \mathbf{v} \in V \mid \mathbf{v} + (-\mathbf{v}) = 0$
Distributivity for vector addition	$a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
Distributivity for field addition	$(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$
Scalar and field multiplication	$a(b\mathbf{v}) = (ab)\mathbf{v}$
Scalar identity element	$1\mathbf{v} = \mathbf{v}$

Table 2.1: The definition of a vector space (Wikipedia 2015)

and language interoperability. A discussion of tradeoffs involved in language choice can be found in (Knepley 2012).

Example: Cartesian vectors We can take arrays of real numbers, indicating extent in the coordinate directions, as our vectors. Addition then means adding the numbers pairwise for each coordinates, and scalar multiplication just means multiplying each entry by the same scalar. We call this space $\mathbb{R}^n$ if there are n coordinates.

Example: Cartesian matrices We take matrices of m rows and n columns with real entries as our vectors. Addition adds the matrices entrywise, and scalar multiplication multiplies each entry. This just reproduces $\mathbb{R}^{mn}$, and corresponds to unrolling a matrix into a vector of length mn.

Example: Polynomials Let us consider polynomials of degree k with real coefficients. We will take as our vectors, each representing a particular polynomial, the arrays of $k + 1$ real coefficients. Pointwise addition of polynomial functions then corresponds exactly to addition of the coefficients,

$$\begin{aligned} p_0(x) + p_1(x) &= \left(a_{00} + a_{01}x + \ldots + a_{0k}x^k\right) + \left(a_{10} + a_{11}x + \ldots a_{1k} + x^k\right) \\ &= (a_{00} + a_{10}) + (a_{01} + a_{11})\, x + \ldots + (a_{0k} + \ldots a_{1k})\, x^k. \end{aligned}$$

Multiplication by a scalar just multiplies each coefficient by the same number. Thus I have exactly $\mathbb{R}^k$. Notice that I can replace the coefficients by any field, such as $\mathbb{C}$.

2.0.1 Useful notation

We will indicate the ith entry of a vector $\mathbf{v}$ with v_i, so that

$$\mathbf{v} = \sum_i v_i \hat{\mathbf{e}}_i \tag{2.2}$$

where $\hat{\mathbf{e}}_i$ is the ith basis vector. The *Kronecker delta* function, written δ_{ij} is zero is the two indices are different and unity if they are the same. For example, the entries of the identity matrix I can be expressed as

$$I_{ij} = \delta_{ij}. \tag{2.3}$$

The Kronecker delta is a very useful device for manipulating indices and representing matrices whose only entries are 0 and 1, such as permutation matrices. Another very useful notation is the *Einstein summation notation*. This declares that repeated indices should be summed over. For example,

$$\begin{aligned} I_{ii} &= \sum_i \delta_{ii} \\ &= \sum_i 1 \\ &= \mathrm{Tr}\{I\} \end{aligned}$$

and in general

$$A_{ii} = \mathrm{Tr}\{A\}.$$

A more complex example would be matrix multiplication, so that $C = AB$ could be expressed as

$$C_{ij} = A_{ik}B_{kj}$$

and also

$$A_{ik}B_{ki} = \mathrm{Tr}\{AB\}.$$

2.1 Inner Products, Orthogonality, and Dual Spaces

We can impose some additional structure on our vector space, namely that we can compare angles between vectors. We will define the *inner product* of two vectors as

$$\mathbf{w} \cdot \mathbf{v} = \sum_i w_i v_i. \tag{2.4}$$

However, we will not use this notation very often, since there is a complication for complex vector spaces. Instead, we will connect the idea of the inner product with that of a dual space, arriving at a more compact and useful notation.

We will define the *dual space* $V^\dagger$ as the space of linear functionals on our vector space V. This means the space of linear mappings from V into the field of scalars for our vector space, usually $\mathbb{R}$ or $\mathbb{C}$. According to the famous Riesz Representation Theorem, the space $V^\dagger$ is isomorphic to V, and we can represent

the action of any function $\psi \in V^\dagger$ on a vector $\mathbf{v} \in V$ by the inner product of some vector $\mathbf{w}$ with $\mathbf{v}$, so that

$$\psi(\mathbf{v}) = \bar{\mathbf{w}} \cdot \mathbf{v} = \sum_i \bar{w}_i v_i. \tag{2.5}$$

Now we define the *Hermitian conjugate* of a vector $w^\dagger$ so that

$$\mathbf{w}^\dagger \mathbf{v} = \sum_i \bar{w}_i v_i. \tag{2.6}$$

Thus, the Hermitian conjugate finds the functional represented by that vector. Sometimes people explain this as having "row" and "column" vectors, but this a cumbersome and fragile way to explain things.

If we take the inner product of a vector with itself, we get the square of its length

$$\|\mathbf{v}\| = \sqrt{\mathbf{v}^\dagger \mathbf{v}} \tag{2.7}$$

which is also the 2-norm of the vector, discussed later on. With this, we can define the angle α between two vectors as

$$\cos\alpha = \frac{\mathbf{w}^\dagger \mathbf{v}}{\|\mathbf{v}\|\|\mathbf{w}\|}. \tag{2.8}$$

Clearly, vectors whose inner product is zero correspond to $\alpha = \pi/2$ or a right angle. We call these vectors *orthogonal*. The most important use of orthogonality is to form bases for the span of a set of linearly independent vectors.

2.2 Bases

A *linear combination* of vectors is the sum

$$\alpha_0 \mathbf{v}_0 + \alpha_1 \mathbf{v}_1 + \cdots + \alpha_n \mathbf{v}_n = \sum_{i=0}^{n} \alpha_i \mathbf{v}_i \tag{2.9}$$

where each α_i is a scalar from some field. The only fields we will use in this class are the real numbers $\mathbb{R}$ and the complex numbers $\mathbb{C}$. The *span* of a set of vectors $\{\mathbf{v}_i\}$ is the subspace of vectors which can be constructed as linear combinations of $\{\mathbf{v}_i\}$. In the quantum mechanics literature, a linear combination of states is called a *superposition*. A *linearly independent* set is a set of vectors where the only linear combination that vanishes, namely

$$\lambda_0 \mathbf{v}_0 + \lambda_1 \mathbf{v}_1 + \cdots \lambda_n \mathbf{v}_n = 0, \tag{2.10}$$

requires that

$$\lambda_0 = \lambda_1 = \cdots = \lambda_n = 0. \tag{2.11}$$

Thus no combination of linearly independent vectors can add up to zero.

A basis $\{\hat{\mathbf{e}}_i\}$ for a vector space V is a set of linearly independent vectors whose span is the entire space, such that every vector $\mathbf{v} \in V$ can be written as a finite linear combination of $\hat{\mathbf{e}}_i$ in a unique way. I am ignoring subtleties connected with infinite dimensional vector spaces, as I will for this entire course. A simple procedure to construct a basis is to start with one vector, which is a trivial linearly independent set. If that vector spans the space, we are done. If not, add a vector which is not in its span, and repeat. If V is finite dimensional, then this process is guaranteed to terminate in d steps, where d is the dimension of the space.

Given that every vector $\mathbf{v} \in V$ can be expressed as a linear combination of basis vectors

$$\mathbf{v} = \sum_i v_i \hat{\mathbf{e}}_i,$$

how do we find the scalars v_i? We can take the inner product of the equation above with some basis vector $\hat{\mathbf{e}}_k$, so that

$$\hat{\mathbf{e}}_k^\dagger \mathbf{v} = \sum_i v_i \hat{\mathbf{e}}_k^\dagger \hat{\mathbf{e}}_i, \tag{2.12}$$

$$\bar{v}_k = \sum_i v_i e_{ki}, \tag{2.13}$$

where we defined $\bar{v}_k = \hat{\mathbf{e}}_k^\dagger \mathbf{v}$ and $e_{ki} = \hat{\mathbf{e}}_k^\dagger \hat{\mathbf{e}}_i$, both of which may be calculated if we know the basis and vector. This may be recast in linear algebraic notation as a matrix equation

$$E\mathbf{v} = \bar{\mathbf{v}} \tag{2.14}$$

$$\mathbf{v} = E^{-1}\bar{\mathbf{v}} \tag{2.15}$$

where $\mathbf{v}$ is the vector of coefficients v_i, $\bar{\mathbf{v}}$ is the vector of coefficients $\bar{v}_i$, and E is the matrix of coefficients e_{ij}. Now if we have an orthonormal basis, which we will assume from here on, then

$$e_{ij} = \hat{\mathbf{e}}_i^\dagger \hat{\mathbf{e}}_k = \delta_{ij}, \tag{2.16}$$

which means that

$$v_i = \bar{v}_i = \hat{\mathbf{e}}_i^\dagger \mathbf{v}. \tag{2.17}$$

Suppose instead that we have two different bases, $\{\hat{\mathbf{e}}_i\}$ and $\{\hat{\mathbf{f}}_i\}$. How would we get the components v_i^f of some vector $\mathbf{v}$ in the f-basis if we already know the compnents v_i^e in the e-basis? This is a practical problem, since we often measure in some basis but do calculations in another. We can derive an expression for this by expanding basis vectors of the first set in terms of basis vector in the

second

$$\mathbf{v} = \sum_i v_i^e \hat{\mathbf{e}}_i \tag{2.18}$$

$$\sum_i v_i^f \mathbf{f}_i = \sum_i v_i^e \hat{\mathbf{e}}_i \tag{2.19}$$

$$\sum_i v_i^f \mathbf{f}_i = \sum_i v_i^e \sum_j V_{ji} \mathbf{f}_j \tag{2.20}$$

where $V_{ji} = \mathbf{f}_j \cdot \hat{\mathbf{e}}_i$ is the expression of the basis vector $\hat{\mathbf{e}}_i$ in the f-basis. Now we can take the dot product with $\mathbf{f}_k$,

$$\sum_i v_i^f \mathbf{f}_k \cdot \mathbf{f}_i = \sum_i v_i^e \sum_j V_{ji} \mathbf{f}_k \cdot \mathbf{f}_j, \tag{2.21}$$

$$\sum_i v_i^f \delta_{ki} = \sum_i v_i^e \sum_j V_{ji} \delta_{kj}, \tag{2.22}$$

$$v_k^f = \sum_i v_i^e V_{ki}, \tag{2.23}$$

$$\mathbf{v}^f = V \mathbf{v}^e. \tag{2.24}$$

Thus the coefficients in the f-basis can be obtained from the coefficients in the e-basis by applying the matrix V with coefficients

$$V_{ij} = \mathbf{f}_i \cdot \hat{\mathbf{e}}_j \tag{2.25}$$

which we will call the *Vandermonde matrix*, although Vandermonde was original talking about a very specific change of basis (see Problem 2).

2.3 Linear Operators

Matrix multiplication is simply the application of a linear operator A between two vector spaces, to an input vector $\mathbf{x}$, generating an output vector $\mathbf{y}$,

$$A\mathbf{x} = \mathbf{y}. \tag{2.26}$$

This operation is required to be linear, namely

$$A(\alpha\mathbf{x} + \beta\mathbf{z}) = \alpha(A\mathbf{x}) + \beta(A\mathbf{z}). \tag{2.27}$$

If we expand the vectors $\mathbf{x}$ and $\mathbf{y}$ in some basis $\{\hat{\mathbf{e}}\}$, noting that a basis is guaranteed to exist for any Hilbert space, we have

$$x = \sum_j x_j \hat{\mathbf{e}}_j \quad \text{and} \quad y = \sum_j y_j \hat{\mathbf{e}}_j, \tag{2.28}$$

and plugging into Eq. (2.26) gives

$$\begin{aligned} A\sum_j x_j\hat{\mathbf{e}}_j &= \sum_j y_j\hat{\mathbf{e}}_j, \\ \sum_j x_j(A\hat{\mathbf{e}}_j) &= \sum_j y_j\hat{\mathbf{e}}_j. \end{aligned} \tag{2.29}$$

Now we suppose that our basis is orthonormal, meaning that

$$\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_j = \delta_{ij}. \tag{2.30}$$

We can take the inner product of Eq. 2.29 with $\hat{\mathbf{e}}_i$,

$$\begin{aligned} \hat{\mathbf{e}}_i \cdot \sum_j x_j(A\hat{\mathbf{e}}_j) &= \hat{\mathbf{e}}_i \cdot \sum_j y_j\hat{\mathbf{e}}_j, \\ \sum_j x_j\hat{\mathbf{e}}_i \cdot (A\hat{\mathbf{e}}_j) &= \sum_j y_j\hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_j, \\ \sum_j x_j\hat{\mathbf{e}}_i \cdot (A\hat{\mathbf{e}}_j) &= \sum_j y_j\delta_{ij}, \\ \sum_j a_{ij}x_j &= y_i. \end{aligned} \tag{2.31}$$

where used the linearity of the inner product in line 2, the orthongonality of basis vectors from Eq. 2.30, and we defined the matrix elements

$$a_{ij} = \hat{\mathbf{e}}_i \cdot (A\hat{\mathbf{e}}_j). \tag{2.32}$$

We see that Eq. 2.31 is exactly our rule for matrix multiplication, but we have derived it from the properties of abstract linear operators and bases. This means that we can use our insights in domains others than the Euclidean space $\mathbb{R}^n$, such as vector spaces of functions.

We will define the Hermitian conjugate, or *adjoint*, $A^\dagger$ of the operator A such that

$$\left(w^\dagger Av\right)^\dagger = v^\dagger A^\dagger w. \tag{2.33}$$

Since we have defined the matrix element $a_{ij} = \hat{\mathbf{e}}_i^\dagger A\hat{\mathbf{e}}_j$, it means that the matrix element $a_{ij}^\dagger$ for the Hermitian conjugate $A^\dagger$ should be

$$a_{ij}^\dagger = \hat{\mathbf{e}}_i^\dagger A^\dagger\hat{\mathbf{e}}_j = \left(\hat{\mathbf{e}}_j^\dagger A\hat{\mathbf{e}}_i\right)^\dagger = a_{ji}^\dagger = \bar{a}_{ji}. \tag{2.34}$$

Thus we get the Hermitian conjugate of a matrix by interchanging rows and columns and taking the complex conjugate. If the matrix is real, then we call this the *transpose*. Suppose that we want the Hermitian conjugate of the product

of two matrices AB, then

$$(AB)^\dagger_{ij} = \left(\sum_k a_{ik}b_{kj}\right)^\dagger \tag{2.35}$$
$$= \sum_k \bar{a}_{jk}\bar{b}_{ki} \tag{2.36}$$
$$= \sum_k B^\dagger_{ik}A^\dagger_{kj} \tag{2.37}$$
$$= \left(B^\dagger A^\dagger\right)_{ij} \tag{2.38}$$

so that $AB^\dagger = B^\dagger A^\dagger$. A similar thing can be proved for inverses, in a simpler way,

$$I = AB\,(AB)^{-1} \tag{2.39}$$
$$= ABB^{-1}A^{-1} \tag{2.40}$$
$$= AA^{-1} \tag{2.41}$$
$$= I. \tag{2.42}$$

2.3.1 Unitary operators

A *unitary* operator U is defined by

$$UU^\dagger = U^\dagger U = I. \tag{2.43}$$

Notice that this implies that the columns of U are orthonormal, since

$$\left(UU^\dagger\right)_{ij} = \sum_k u_{ik}\bar{u}_{jk} \tag{2.44}$$
$$= u_i \cdot u_j^\dagger \tag{2.45}$$
$$= \delta_{ij} \tag{2.46}$$

where u_i is the ith column of U. A unitary transformation is an *isometry*, meaning a transformation which preserves the matric on a space or the norm of every vector. More precisely, unitary operators are L_2 isometries because they preserve the 2-norm of vectors,

$$\|Ux\|_2 = (Ux)^\dagger(Ux) \tag{2.47}$$
$$= x^\dagger U^\dagger U x \tag{2.48}$$
$$= x^\dagger x \tag{2.49}$$
$$= \|x\|_2. \tag{2.50}$$

A very common type of unitary operator is a permutation matrix, which a single one in each row. The row represents the new index and the column the

old index. Since applying the permutation followed by the inverse permutation gives the identity, it is unitary. We can show this by using the Kronecker delta to express the elements of a permutation matrix P,

$$P_{ij} = \delta_{i\sigma(j)} \tag{2.51}$$

where $\sigma(k)$ is the permutation function, giving the index for element k after permutation. We can see this by acting on the basis vector $\hat{\mathbf{e}}_k$ with P,

$$\begin{aligned}(P\hat{\mathbf{e}}_k)_i &= \sum_j P_{ij}(\hat{\mathbf{e}}_k)_j && (2.52)\\ &= \sum_j \delta_{i\sigma(j)}\delta_{kj} && (2.53)\\ &= \delta_{i\sigma(k)} && (2.54)\end{aligned}$$

so that the output is $\hat{\mathbf{e}}_{\sigma(k)}$. Now we can look at the matrix product

$$\begin{aligned}\left(P^\dagger P\right)_{ij} &= \sum_k P^\dagger_{ik}P_{kj} && (2.55)\\ &= \sum_k P_{ki}P_{kj} && (2.56)\\ &= \sum_k \delta_{k\sigma(i)}\delta_{k\sigma(j)} && (2.57)\\ &= \delta_{\sigma(i)\sigma(j)} && (2.58)\\ &= \delta_{ij} && (2.59)\end{aligned}$$

where the last step follows because σ is one-to-one. Thus $P^\dagger P = I$ and P is unitary.

2.3.2 Block Matrices

As a consequence of linearity, we can simplify the presentation of matrices with block structure. Consider the 4×4 matrix and vector

$$A = \left(\begin{array}{cc|cc} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ \hline a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{array}\right) \qquad x = \left(\begin{array}{c} x_0 \\ x_1 \\ \hline x_2 \\ x_3 \end{array}\right).$$

The expression for Ax is given by

$$(Ax)_i = \sum_j a_{ij}x_j,$$

but we can express the loop over $j \in [0, 4)$ as two loops by splitting the index into subindices $j = k * 2 + l$ for $k, l \in [0, 2)$,

$$\begin{aligned}(Ax)_i &= \sum_k \sum_l a_{i,k*2+l} x_{k*2+l}, \\ &= \sum_k \sum_l a_{i,(kl)} x_{(kl)},\end{aligned}$$

where (kl) is multindex, k indicating which block and l the index within that block. Now suppose we also index the output vector using our multindex (ij), so that

$$\begin{aligned}(Ax)_{(ij)} &= \sum_k \sum_l a_{(ij),(kl)} x_{(kl)}, \\ (Ax)_i &= \sum_k a_{i,k} x_k,\end{aligned}$$

where

$$a_{i,k} x_k = \begin{pmatrix} a_{(i0),(k0)} & a_{(i0),(k1)} \\ a_{(i1),(k0)} & a_{(i1),(k1)} \end{pmatrix} \begin{pmatrix} x_{(k0)} \\ x_{(k1)} \end{pmatrix}$$

so that our usual rule for matrix-vector multiplication applies to the individual blocks, and we can write

$$A = \left(\begin{array}{c|c} a_{00} & a_{01} \\ \hline a_{10} & a_{11} \end{array}\right) \qquad x = \left(\begin{array}{c} x_0 \\ \hline x_1 \end{array}\right),$$

where each entry is a small vector or matrix, and multiplication is understood to be matrix-vector multiplication. This same procedure extends to matrix-matrix multiplication, and on to more general tensors.

2.4 Tensor Product Spaces

The *tensor product* $V \otimes W$ of two vector spaces V and W (over the same field) is itself a vector space, together with an operation of bilinear composition, denoted by $\otimes$, from ordered pairs in the Cartesian product $V \times W$ into $V \otimes W$. The tensor product is defined by the bilinearity of the product operation $\otimes$,

$$\begin{aligned} \forall \mathbf{v} \in V, \forall \mathbf{w}_0, \mathbf{w}_1 \in W \qquad & \mathbf{v} \otimes (\alpha_0 \mathbf{w}_0) + \mathbf{v} \otimes (\alpha_1 \mathbf{w}_1) = \mathbf{v} \otimes (\alpha_0 \mathbf{w}_0 + \alpha_1 \mathbf{w}_1), \\ \forall \mathbf{v}_0, \mathbf{v}_1 \in V, \forall \mathbf{w} \in W \qquad & (\alpha_0 \mathbf{v}_0) \otimes \mathbf{w} + (\alpha_1 \mathbf{v}_1) \otimes \mathbf{w} = (\alpha_0 \mathbf{v}_0 + \alpha_1 \mathbf{v}_1) \otimes \mathbf{w}. \end{aligned}$$

Given two linear operators $A : V \to X$ and $B : W \to Y$, we define the tensor product of the operators as a linear map

$$A \otimes B : V \otimes W \to X \otimes Y \tag{2.60}$$

such that

$$(A \otimes B)(\mathbf{v} \otimes \mathbf{w}) = (A\mathbf{v}) \otimes (B\mathbf{w}), \tag{2.61}$$

which also implies that

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{2.62}$$

We can get the action of the combined operator on a combined vector by inserting bases for the two spaces $\{\hat{\mathbf{e}}_i\}$ and $\{\hat{\mathbf{f}}_i\}$,

$$(A \otimes B)(\sum_i v_i \hat{\mathbf{e}}_i \otimes \sum_j w_j \hat{\mathbf{f}}_j) = (A \sum_i v_i \hat{\mathbf{e}}_i) \otimes (B \sum_j w_j \hat{\mathbf{f}}_j), \tag{2.63}$$

$$(A \otimes B)\left(\sum_i \sum_j v_i w_j (\hat{\mathbf{e}}_i \otimes \hat{\mathbf{f}}_j)\right) = (\sum_i v_i A\hat{\mathbf{e}}_i) \otimes (\sum_j w_j B\hat{\mathbf{f}}_j). \tag{2.64}$$

Then we can get a matrix representation of the combined operator if we let the input vector be a tensor product of the basis vectors,

$$(A \otimes B)\left(\hat{\mathbf{e}}_i \otimes \hat{\mathbf{f}}_j\right) = (A\hat{\mathbf{e}}_i) \otimes (B\hat{\mathbf{f}}_j), \tag{2.65}$$

and look at the (kl) entry of the output vector by taking the dot product with that basis vector,

$$\left(\hat{\mathbf{e}}_k \otimes \hat{\mathbf{f}}_l\right)^\dagger (A \otimes B)\left(\hat{\mathbf{e}}_i \otimes \hat{\mathbf{f}}_j\right) = \left(\hat{\mathbf{e}}_k \otimes \hat{\mathbf{f}}_l\right)^\dagger \left((A\hat{\mathbf{e}}_i) \otimes (B\hat{\mathbf{f}}_j)\right), \tag{2.66}$$

$$(A \otimes B)_{(kl),(ij)} = (\hat{\mathbf{e}}_k^\dagger A\hat{\mathbf{e}}_i)(\hat{\mathbf{f}}_l^\dagger B\hat{\mathbf{f}}_j), \tag{2.67}$$

$$= a_{ki} b_{lj} \tag{2.68}$$

which is precisely the *Kronecker product* of matrices A and B, defined here. Notice that we have made a block matrix of exactly the type we saw in Section 2.3.2. For example, the Krocker product of two 2×2 matrices is given by

$$\begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix} \otimes \begin{pmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix} \tag{2.69}$$

$$= \begin{pmatrix} a_{0,0}\begin{pmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix} & a_{0,1}\begin{pmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix} \\ a_{1,0}\begin{pmatrix} b_{0,1} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix} & a_{1,1}\begin{pmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{pmatrix} \end{pmatrix} \tag{2.70}$$

$$= \begin{pmatrix} a_{0,0}b_{0,0} & a_{0,0}b_{0,1} & a_{0,1}b_{0,0} & a_{0,1}b_{0,1} \\ a_{0,0}b_{1,0} & a_{0,0}b_{1,1} & a_{0,1}b_{1,0} & a_{0,1}b_{1,1} \\ a_{1,0}b_{0,0} & a_{1,0}b_{0,1} & a_{1,1}b_{0,0} & a_{1,1}b_{0,1} \\ a_{1,0}b_{1,0} & a_{1,0}b_{1,1} & a_{1,1}b_{1,0} & a_{1,1}b_{1,1} \end{pmatrix}. \tag{2.71}$$

In all of our quantum computing examples, we will be looking at combinations of 2-state quantum systems, so that all our tensor product operators will look

like this. Note that A is indexed with the high bit and B the low bit. If we have a tensor product of several 2×2 operators, then each one will be indexed by a given bit of the global index.

Since the action of tensor product operators can be decomposed into action on separate spaces, we can establish useful theorems about them. For example, using our definition above for adjoints, we see that

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger. \tag{2.72}$$

Suppose that we have the tensor product of two unitary operators. Is it also unitary? We can prove this using Eq. (2.72),

$$\begin{aligned}
(U_1 \otimes U_1)^\dagger (U_1 \otimes U_1) &= \left(U_1^\dagger \otimes U_2^\dagger\right)(U_1 \otimes U_1) && (2.73)\\
&= \left(U_1^\dagger U_1 \otimes U_2^\dagger U_2\right) && (2.74)\\
&= (I \otimes I) && (2.75)\\
&= I. && (2.76)
\end{aligned}$$

2.5 Norms

2.6 SVD

$$\begin{aligned}
\|A\|_F^2 &= \mathrm{Tr}\left(A^\dagger A\right) && (2.77)\\
&= \mathrm{Tr}\left(\left(U\Sigma V^\dagger\right)^\dagger U\Sigma V^\dagger\right) && (2.78)\\
&= \mathrm{Tr}\left(V\Sigma^\dagger U^\dagger U\Sigma V^\dagger\right) && (2.79)\\
&= \mathrm{Tr}\left(V\Sigma^\dagger\Sigma V^\dagger\right) && (2.80)\\
&= \mathrm{Tr}\left(\Sigma^\dagger\Sigma V^\dagger V\right) && (2.81)\\
&= \mathrm{Tr}\left(\Sigma\Sigma\right) && (2.82)\\
&= \sum_i \sigma_i^2 && (2.83)
\end{aligned}$$

2.7 Eigenproblems

2.8 Problems

Problem II.1 This problem will familiarize you with the grading system that we use at UB. Follow the steps below to ensure that your Autolab account is working correctly.

1. Create your account at https://autograder.cse.buffalo.edu using your UB email address.

2. An account may have been created for you if you enrolled before you had an account If Autolab says that you already have an account, click "Forgot your password?" and enter your email address. Follow instructions to reset your password.

3. Ensure that you are registered for the course: CSE410: Quantum Algorithms (Fall 19)

4. Submit a pdf to Homework 0 with the following information:

   - Name
   - Person number
   - The equation

$$\frac{x^T Ax}{||x||^2} \to \lambda_{\max}$$

   since equation writing will be essential in this course.

The best way to create PDF from LaTeX is to use `pdflatex`,

```
pdflatex essay.tex
bibtex essay
pdflatex essay.tex
pdflatex essay.tex
```

where the repetition is necessary to assure that the metadata stored in auxiliary files is consistent. This process can be handled in an elegant way by using the `latexmk` program,

```
latexmk -pdf essay.tex
```

If you rely on TeX source or BibTeX files in other locations, you can use

```
TEXINPUTS=${TEXINPUTS}:/path/to/tex BIBINPUTS=${BIBINPUTS}:/path/to/bib
  latexmk -pdf essay.tex
```

Problem II.2 Show that the original Vandermonde matrix is actually a change of basis from the monomial basis $\{x^k\}$ to the basis of point evaluation functionals $\{\eta_{x_i}\}$

$$\eta_z(\phi) = \int \phi(x)\delta(x-z)dx \tag{2.84}$$

Problem II.3 Implement both Classical and Modified Gram-Schmidt orthogonalization in PETSc. Use an example to show instability in the classical algorithm that is not present in the modified form.

Problem II.4 NLA 1.1

Problem II.5 NLA 1.3

Problem II.6 NLA 1.4

Problem II.7 NLA 2.1

Problem II.8 NLA 2.2

Problem II.9 NLA 2.3

Problem II.10 NLA 2.4

Problem II.11 NLA 2.6

Problem II.12 NLA 2.7

Problem II.13 NLA 3.1

Problem II.14 NLA 3.3

Problem II.15 NLA 3.6

Problem II.16 NLA 4.1

Problem II.17 NLA 4.4

Problem II.18 NLA 5.3

Problem II.19 NLA 5.4

Problem II.20 QALA 3.1

Problem II.21 QALA 3.4

Problem II.22 QALA 3.5

Problem II.23 QALA 3.7

Problem II.24 QALA 3.9

Problem II.25 QALA 3.10

Problem II.26 QALA 3.12

Problem II.27 QALA 3.16

Problem II.28 QALA 3.17

Problem II.29 NLA 6.1

Problem II.30 NLA 6.3

Problem II.31 NLA 6.5

Problem II.32 NLA 7.1

Problem II.33 NLA 7.3

Problem II.34 NLA 7.4

Problem II.35 NLA 12.2

Problem II.36 NLA 14.1

Problem II.37 NLA 15.2

References

Fearnley-Sander, Desmond (1979). "Hermann Grassmann and the Creation of Linear Algebra". In: *The American Mathematical Monthly* 86, pp. 809–817. URL: http://www.maa.org/sites/default/files/pdf/upload_library/22/Ford/DesmondFearnleySander.pdf.

Wikipedia (2015). *Linear Algebra*. https://en.wikipedia.org/wiki/Linear_algebra. URL: https://en.wikipedia.org/wiki/Linear_algebra.

Lawson, C. L., R. J. Hanson, D. Kincaid, and F. T. Krogh (1979). "Basic linear algebra subprograms for Fortran usage". In: *ACM Transactions on Mathematical Software* 5, pp. 308–323.

Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen (May 1990). *LAPACK: A portable linear algebra library for high-performance computers*. Tech. rep. CS-90-105. Computer Science Dept., University of Tennessee.

Falgout, R. (2017). *hypre Users Manual*. Tech. rep. Revision 2.11.2. Lawrence Livermore National Laboratory.

— (n.d.). *hypre Web page*. http://www.llnl.gov/CASC/hypre.

Heroux, Michael A. and James M. Willenbring (2003). *Trilinos Users Guide*. Tech. rep. SAND2003-2952. Sandia National Laboratories. URL: http://trilinos.sandia.gov/.

Heroux et al., M. (n.d.). *Trilinos Web page*. `http://trilinos.sandia.gov/`.

Bastian, Peter, Markus Blatt, Andreas Dedner, Christian Engwer, Jorrit Fahlke, Christoph Gersbacher, Carsten Gräser, Christoph Grüninger Robert Klöfkorn, Steffen Müthing, Martin Nolte, Mario Ohlberger, and Oliver Sander (2015). *DUNE Web page*. http://www.dune-project.org/. URL: http://www.dune-project.org/.

Jacob, Benoit and Gaël Guennebaud (2015). *Eigen Web page*. http://eigen.tuxfamily.org/. URL: http://eigen.tuxfamily.org/.

Poulson, Jack, Bryan Marker, Jeff R. Hammond, Nichols A. Romero, and Robert van de Geijn (2013). "Elemental: A New Framework for Distributed Memory Dense Matrix Computations". In: *ACM Transactions on Mathematical Software* 39.2.

Poulson, Jack (2015). *Elemental: Distributed memory dense linear algebra*. http://libelemental.org. URL: http://libelemental.org/.

Balay, Satish, Shrirang Abhyankar, et al. (2020). *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.13. Argonne National Laboratory.

— (2019). *PETSc Web page*. https://www.mcs.anl.gov/petsc. URL: https://www.mcs.anl.gov/petsc.

Balay, Satish, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith (1997). "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries". In: *Modern Software Tools in Scientific Computing*. Ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen. Birkhauser Press, pp. 163–202.

Knepley, Matthew G. (2012). "Programming Languages for Scientific Computing". In: *Encyclopedia of Applied and Computational Mathematics*. Ed. by Björn Engquist. http://arxiv.org/abs/1209.1711. Springer. DOI: 10.1007/978-3-540-70529-1. URL: http://arxiv.org/abs/1209.1711.

The central difference between classical and quantum spaces is that there are N pure states for N degrees of freedom in a classical system, whereas a quantum system can support an infinite number of pure states and has dimension N^2. Since there are N degrees of freedom, we have N basis vectors in our space, classical or quantum. Since we will always construct our composite systems from a combination of two-state systems, our full system will be a tensor product and $N = 2^n$ where n is the number of two-state systems. This also means that we can make a correspondence between n-bit Boolean strings and basis vectors. For example, a basis vector of the full system $\hat{\mathbf{e}}_M$ where the number M can be expressed by the bit string

$$M = m_0 m_1 \dots m_n,$$

is given by the product of basis vectors of the two-state subsystems

$$\hat{\mathbf{e}}_M = \hat{\mathbf{e}}_{m_0} \otimes \hat{\mathbf{e}}_{m_1} \otimes \cdots \otimes \hat{\mathbf{e}}_{m_n}.$$

Chapter 3

Boolean and Hilbert Spaces

Quantum computers are machines for manipulating qubits. A *qubit*, or quantum bit, is the basic unit of quantum information, the quantum mechanical version of the classical binary bit. A qubit is a two-state quantum mechanical system, such as the spin of the electron in which the two states can be taken as spin up and spin down, or the polarization of a single photon in which the two states can be taken to be the vertical polarization and the horizontal polarization. In a classical system, a bit would have to be either true or false. However, quantum mechanics allows the qubit to be in a coherent superposition of both states at the same time, a property that is fundamental to quantum mechanics and thus quantum computing. In terms of probabilities, as pointed out by Hardy, quantum amplitudes can be negative, and lead to cancellation, whereas classical probabilities must be positive. Therefore a qubit corresponds to a line in a quantum circuit diagram, but not to a row of the permutation matrix representing our invertible function F. The state space for a full problem is a tensor product of individual spaces for each qubit. When quantum mechanics refer to a *superposition* of states, what they mean is that we have a linear combination of tensor product basis vectors.

3.1 Boolean Functions

In order to talk about two-state quantum systems, we will use the language of boolean functions, where we identify the two quantum states with T and F. A *unary Boolean function* operates on a single bit and returns a single bit. There are only two unary functions, NOT and the identity. A *binary Boolean function* operates on two input bits and returns a single output bit. These are the familiar functions, such as AND and OR. We can generalize Boolean functions to strings of bits in at least two different ways. A *bitwise Boolean function* applies a binary function to each pair of bits and collects the output bits into another string, whereas an *n-ary Boolean function* operates on all argument bits to produce a single output bit. For example, n-ary AND returns

T only if all input bits are T, and n-ary OR returns F only if all inputs are F. The n-ary XOR function returns T only if an odd number of argument bits are T, which makes sense given its identification with addition modulo 2. We can define the *Boolean inner product* of two bit strings as n-ary XOR of the bitwise AND of the two input strings,

$$x \cdot y = x_1 y_1 \oplus \cdots \oplus x_m y_m. \tag{3.1}$$

This makes some sense in that n-ary XOR looks like addition and bitwise AND looks like multiplication, and we retain the distributive property,

x	y	z	$x \wedge (y \oplus z)$	$(x \wedge y) \oplus (x \wedge z)$
T	T	T	F	F
T	T	F	T	T
T	F	T	T	T
T	F	F	F	F
F	T	T	F	F
F	T	F	F	F
F	F	T	F	F
F	F	F	F	F

Note that n-ary XOR is addition mod 2 of the input bits, rather than addition mod 2 of the input numbers represented by the bit strings. The table for $x \oplus y$ where x and y are single bits is given by

y \ x	0	1
0	0	1
1	1	0

If we instead look at two bit strings,

y \ x	00	01	10	11
00	0	1	0	1
01	1	0	1	0
10	0	1	0	1
11	1	0	1	0

we have copies of the first table, because only the least significant bits matter in addition modulo two.

3.2 Matrix Representations

If we imagine a system composed of n two-state quantum systems, the size of the overall Hilbert space for the combined system is $N = 2^n$, because it is the tensor product of two-dimensional spaces. We can label each basis function of the combined system by its number in binary, so the rightmost bit is fastest. For example, if we combine two electrons, we have states 00, 01, 10, and 11 where 0 and 1 correspond to spin up and spin down basis states.

In quantum mechanics, linear operators transform basis states into each other, and in fact we require that the operators be unitary in closed systems so that the total probability for all measurements remains unity. This mapping can be seen as a transformation of truth values from input to output. However, if we want to represent unitary mappings, we must use invertible Boolean functions. The Boolean function $f(x_1, \ldots, x_n) = y$ is not invertible, so instead we create an invertible function F from it

$$F(x_1, \ldots, x_n, z) = (x_1, \ldots, x_n, z \oplus f(x_1, \ldots, x_n)) \tag{3.2}$$

which can be shown to be its own inverse

$$\begin{aligned} F\left(F(x_1, \ldots, x_n, z)\right) &= F\left(x_1, \ldots, x_n, z \oplus f(x_1, \ldots, x_n)\right) \\ &= (x_1, \ldots, x_n, (z \oplus f(x_1, \ldots, x_n)) \oplus f(x_1, \ldots, x_n)) \\ &= (x_1, \ldots, x_n, z)\,. \end{aligned}$$

As a simple example, the unary NOT function is invertible, and has the representation

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.3}$$

We can check that it is invertible (and unitary) by noting that $X^2 = I$. Now suppose we use our strategy for making invertible Boolean functions on the identity function so that we have $f(x) = x$, the we use $F(x_1, x_2) = (x_1, x_2 \oplus x_!)$. Then we have the matrix representation

$$\begin{array}{c} \\ e_{00} \\ e_{01} \\ e_{10} \\ e_{11} \end{array} \begin{array}{c} \begin{array}{cccc} e_{00} & e_{01} & e_{10} & e_{11} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}. \tag{3.4}$$

We will call this operation **CNOT** (controlled NOT), since it negates the second argument if and only if the first argument is T. Thus the first argument is *controlling* the NOT on the second. We also note that this matrix cannot be written as a Kronecker product of simpler gates.

In general, the matrix associated with F is size $2N \times 2N$, since we add an extra argument z. We can label each row by the input $x_1 x_2 \cdots x_n z$, and each one will have only a single nonzero in column $x_1 x_2 \cdots x_n (z \oplus f(x_1, \ldots, x_n))$. A matrix with this structure is a *permutation matrix*, which we denote $\boldsymbol{P}_f$. Note that permutation matrices are unitary, which implies the invertibility of F. We can keep going in this fashion by making extra inputs $z_1, \ldots, z_m$ if the function f has m outputs.

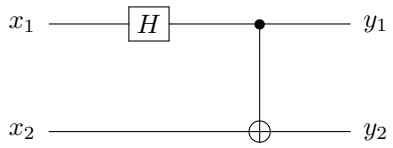
Rules for Feasibility:

1. Any unitary operator $\boldsymbol{B}$ of size 2^k for fixed k is feasible. These are operations involving a fixed number k of qubits.

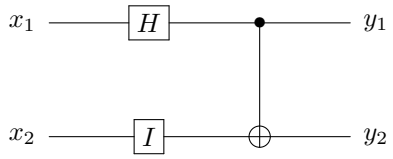
2. A tensor product of $\boldsymbol{B}$ with identity matrices is feasible. We will call this a *basic operator*. Note that this is also unitary.

3. The multiplication $\boldsymbol{U}_1 \cdots \boldsymbol{U}_t$ of a polynomial number of feasible operators, so that $t = n^{\mathcal{O}(1)}$, is feasible. We can generalize this to allow $s = n^{\mathcal{O}(1)}$ qubits instead of just n qubits.

Lets look at a simple quantum circuit



which acts on qubit 1 with a Hadamard gate and then feeds both qubits into a **CNOT**. There is an implied identity transformation on qubit 2, which could be included explicitly



in order to make the transition to linear algebra clearer. We can get out linear algebraic form U for this circuit

$$
\begin{aligned}
U &= U_2 U_1 && (3.5)\\
&= \mathbf{CNOT}\,(\mathbf{H} \otimes \mathbf{I}) && (3.6)\\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) && (3.7)\\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} && (3.8)\\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} && (3.9)\\
& && (3.10)
\end{aligned}
$$

Suppose that we act on the input state e_{00} which means that both qubits are

in the spin up or F state,

$$Ue_{00} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.11}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.12}$$

$$= \frac{1}{\sqrt{2}} \left(e_{00} + e_{11} \right). \tag{3.13}$$

The output state is the so-called *Bell state*, meaning a maximally entangled state, since if I measure the first qubit and get 0 I know immediately that the other quibit must be 0, and likewise with 1. Thus the circuit above is routinely used to construct an entangled pair from a simple initial state.

3.3 Problems

Problem III.1 QALA 2.1

Problem III.2 QALA 2.3

Problem III.3 QALA 2.4

Problem III.4 QALA 2.5

Problem III.5 QALA 2.6

Problem III.6 QALA 2.7

Problem III.7 QALA 2.8

Problem III.8 QALA 4.2

Problem III.9 QALA 4.3

Problem III.10 QALA 4.4

Problem III.11 QALA 4.5

Problem III.12 QALA 4.6

Problem III.13 QALA 4.8

Problem III.14 QALA 4.9

Problem III.15 QALA 4.10

Problem III.16 QALA 4.12

Problem III.17 QALA 4.13

Problem III.18 QALA 4.14

Problem III.19 QALA 6.1

Problem III.20 QALA 6.2

Problem III.21 QALA 6.4

Problem III.22 QALA 6.5

Problem III.23 QALA 6.6

Problem III.24 QALA 6.7

Problem III.25 QALA 6.8

Problem III.26 QALA 6.9

Problem III.27 QALA 6.10

Chapter 4

Quantum Algorithms

The bag of tricks for quantum computing, I think, arises mainly, not from quantum properties, but rather from the necessity of reversibility. This is the origin, for instance, of the famous No-Cloning Theorem of quantum mechanics. This says that there does not exist a universal unitary transformation which produce an exact copy an unknown quantum state. To be specific, let us define a state a, and ask for a transformation U_C such that

$$U_C(a \otimes e_0) = e^{i\alpha(a,e_0)}(a \otimes a)$$

Now suppose that we cloned two states a and b, and looked at their inner product

$$\begin{aligned}(b \otimes e_0)^\dagger(a \otimes e_0) &= (b \otimes e_0)^\dagger U_C^\dagger U_C(a \otimes e_0)\\ (b^\dagger a)(e_0^\dagger e_0) &= (U_C(b \otimes e_0))^\dagger (U_C(a \otimes e_0))\\ (b^\dagger a) &= e^{-i\alpha(b,e_0)}(b \otimes b)^\dagger e^{i\alpha(a,e_0)}(a \otimes a)\\ |b^\dagger a| &= |b^\dagger a|^2\end{aligned}$$

This implies that either $b^\dagger a = 0$ or $b^\dagger a = 1$. Hence by the Cauchy-Schwarz Inequality the states are either parallel or orthogonal. This cannot be the case for two arbitrary states, and therefore, a single universal U_C cannot clone a general quantum state. Notice that we could design a U to copy a given quantum state, but the transformation would depend on the state it was copying.

We can, however, create a transform U_C that obeys

$$b_{ij} = a'_{i(i\oplus j)}$$

where $\oplus$ is understood as the bitwise operator. If i and j are single bits, this is just the action of **CNOT**, which is our reversibility transform applied to the identity. If we apply **CNOT** to each pair of qubits, we can create this state for $2n$ qubits. Lets look at the effect for a 2-qubit system. Suppose I apply **CNOT**

to $a \otimes e_0$,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ 0 \\ a_1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_0 \\ 0 \\ 0 \\ a_1 \end{pmatrix}$$

so that $b_{ii} = a_i$. This means that for any basis vector, $U_C(e_k \otimes e_0) = e_k \otimes e_k$. However, lets apply this to the state $\frac{1}{\sqrt{2}}(e_0 + e_1) \otimes e_0$,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

but that is not the cloned state

$$\frac{1}{\sqrt{2}}(e_0 + e_1) \otimes \frac{1}{\sqrt{2}}(e_0 + e_1) = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

We can also do the computation symbolically

$$\begin{aligned} U_C \left(\frac{1}{\sqrt{2}}(e_0 + e_1) \otimes e_0 \right) &= \frac{1}{\sqrt{2}} U_C \left(e_0 \otimes e_0 + e_1 \otimes e_0 \right) \\ &= \frac{1}{\sqrt{2}} \left(e_0 \otimes e_0 + e_1 \otimes e_1 \right) \end{aligned}$$

whereas

$$\frac{1}{\sqrt{2}}(e_0 + e_1) \otimes \frac{1}{\sqrt{2}}(e_0 + e_1) = \frac{1}{2}(e_0 \otimes e_0 + e_0 \otimes e_1 + e_1 \otimes e_0 + e_1 \otimes e_1).$$

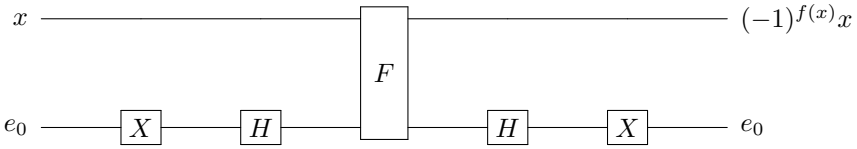
Suppose we want to select out some qubits from among the output of some operation U. Then we just act with $U \otimes I$ on the initial state $a \otimes e_0$. Then pick out the qubits we want using C_m, where the **CNOT** controls are on our chosen qubits, and the targets are on the m ancillary qubits we are using to make things reversible. Then act with $U^\dagger \otimes I$ to return the original inputs. This is called the Copy-Uncompute trick.

By linearity of the tensor product, scaling one part is equivalent to scaling the other parts

$$\alpha(\mathbf{v} \otimes \mathbf{w}) = \alpha\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \otimes \alpha\mathbf{w}$$

We can show this explicitly by calculating the Kronecker product of two vectors,

$$\begin{aligned}
\alpha\left(\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\right) &= \alpha\begin{pmatrix}0\\1\\0\\0\end{pmatrix}\\
&= \begin{pmatrix}0\\\alpha\\0\\0\end{pmatrix}\\
&= \begin{pmatrix}\alpha\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\\
&= \begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\\alpha\end{pmatrix}
\end{aligned}$$



The unitary operator $\mathcal{F}$ corresponding to the boolean function f is linear, so that its action on mixed states is given by the linear combination of the action on pure states. Moreover, it is a permutation, mapping the input state labeled by xz to the output state labeled by $x(x \oplus f(x))$. Thus, we may write

$$\begin{aligned}
F(x,d) &= \mathcal{F}(e_x \otimes e_d)\\
&= \frac{1}{\sqrt{2}}\left(\mathcal{F}(e_x \otimes e_0) - \mathcal{F}(e_x \otimes e_1)\right)\\
&= \frac{1}{\sqrt{2}}\left(F(x0) - F(x1)\right)\\
&= \frac{1}{\sqrt{2}}\left(\left(e_x \otimes e_{0\oplus f(x)}\right) - \left(e_x \otimes e_{1\oplus f(x)}\right)\right)\\
&= \frac{1}{\sqrt{2}}\left(e_x \otimes \left(e_{0\oplus f(x)} - e_{1\oplus f(x)}\right)\right)\\
&= \frac{1}{\sqrt{2}}\left(e_x \otimes (-1)^{f(x)}\left(e_0 - e_1\right)\right)\\
&= (-1)^{f(x)}\left(e_x \otimes \frac{1}{\sqrt{2}}\left(e_0 - e_1\right)\right)\\
&= \left((-1)^{f(x)} e_x \otimes \mathbf{d}\right)
\end{aligned}$$

This shows that the Grover Oracle is feasible, since we can start with $\mathbf{j}$, and then act with this circuit to multiply all entries in the true set S with negative one.

4.1 Examples

4.1.1 Create superposition

Quantum Circuit:

Functional:

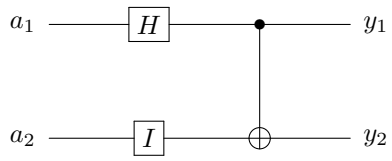
$$\mathbf{y} = \mathbf{H}e_0 = \frac{1}{\sqrt{2}}(e_0 + e_1) \tag{4.1}$$

Linear Algebraic:

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} \tag{4.2}$$

4.1.2 Create entanglement

Quantum Circuit:



Functional:

$$\begin{aligned}
\mathbf{y} &= \mathbf{CNOT}\,(\mathbf{H}\otimes I)\,(e_0\otimes e_0) && (4.3)\\
&= \mathbf{CNOT}\,(\mathbf{H}e_0\otimes e_0) && (4.4)\\
&= \frac{1}{\sqrt{2}}\mathbf{CNOT}\,(e_0\otimes e_0 + e_1\otimes e_0) && (4.5)\\
&= \frac{1}{\sqrt{2}}\mathbf{CNOT}\,(e_{00} + e_{10}) && (4.6)\\
&= \frac{1}{\sqrt{2}}(e_{00} + e_{11}) && (4.7)
\end{aligned}$$

Linear Algebraic:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \tag{4.8}$$

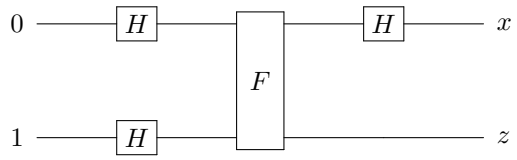
$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \tag{4.9}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{4.10}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.11}$$

4.1.3 Deutsch's Algorithm

Quantum Circuit:



Linear Algebraic:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} U_F \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{4.12}$$

$$= \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} U_F \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \tag{4.13}$$

Now if f is the identity, $F(x,z) = (x, x \oplus z)$, then

$$\frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} U_F \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \tag{4.14}$$

$$= \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \tag{4.15}$$

$$= \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \tag{4.16}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} \tag{4.17}$$

and a measurement when the first qubit is 0 will yield 0. Similarly for negation, $F(x,z) = (x, (\neg x) \oplus z$, we get

$$\frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \tag{4.18}$$

$$= \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \tag{4.19}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \end{pmatrix} \tag{4.20}$$

with the same result. However for f the always true function, $F(x,z) = (x, \neg z)$,

we get

$$\frac{1}{2\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} \tag{4.21}$$

$$=\frac{1}{2\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\begin{pmatrix}-1\\1\\-1\\1\end{pmatrix} \tag{4.22}$$

$$=\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\\0\end{pmatrix} \tag{4.23}$$

and a measurement when the first qubit is 0 will yield either state for the second qubit with probability $1/2$, and it is similar for f the always false function,

$$\frac{1}{2\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} \tag{4.24}$$

$$=\frac{1}{2\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} \tag{4.25}$$

$$=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\\0\\0\end{pmatrix} \tag{4.26}$$

Functional:

$$(\mathbf{H}\otimes I)\,F\,(\mathbf{H}\otimes\mathbf{H})\,(\hat{\mathbf{e}}_0\otimes\hat{\mathbf{e}}_1) \tag{4.27}$$

$$=(\mathbf{H}\otimes I)\,F\frac{1}{2}\left((\hat{\mathbf{e}}_0+\hat{\mathbf{e}}_1)\otimes(\hat{\mathbf{e}}_0-\hat{\mathbf{e}}_1)\right) \tag{4.28}$$

$$=\frac{1}{2}\,(\mathbf{H}\otimes I)\,F\,(\hat{\mathbf{e}}_{00}-\hat{\mathbf{e}}_{01}+\hat{\mathbf{e}}_{10}-\hat{\mathbf{e}}_{11}) \tag{4.29}$$

$$=\frac{1}{2}\,(\mathbf{H}\otimes I)\left(\hat{\mathbf{e}}_{0f(0)}-\hat{\mathbf{e}}_{0\neg f(0)}+\hat{\mathbf{e}}_{1f(1)}-\hat{\mathbf{e}}_{1\neg f(1)}\right) \tag{4.30}$$

$$\begin{aligned}=&\frac{1}{2\sqrt{2}}(\hat{\mathbf{e}}_{0f(0)}+\hat{\mathbf{e}}_{1f(0)}-\hat{\mathbf{e}}_{0\neg f(0)}-\hat{\mathbf{e}}_{1\neg f(0)}+\\&\hat{\mathbf{e}}_{0f(1)}-\hat{\mathbf{e}}_{1f(1)}-\hat{\mathbf{e}}_{0\neg f(1)}+\hat{\mathbf{e}}_{1\neg f(1)})\end{aligned} \tag{4.31}$$

so if f is constant, we have

$$\frac{1}{2\sqrt{2}}\left(\hat{\mathbf{e}}_{0y}+\hat{\mathbf{e}}_{1y}-\hat{\mathbf{e}}_{0\neg y}-\hat{\mathbf{e}}_{1\neg y}+\hat{\mathbf{e}}_{0y}-\hat{\mathbf{e}}_{1y}-\hat{\mathbf{e}}_{0\neg y}+\hat{\mathbf{e}}_{1\neg y}\right) \tag{4.32}$$

$$=\frac{1}{\sqrt{2}}\left(\hat{\mathbf{e}}_{0y}-\hat{\mathbf{e}}_{0\neg y}\right) \tag{4.33}$$

whereas if f is not constant, we have

$$\frac{1}{2\sqrt{2}}\left(\hat{\mathbf{e}}_{0y}+\hat{\mathbf{e}}_{1y}-\hat{\mathbf{e}}_{0\neg y}-\hat{\mathbf{e}}_{1\neg y}+\hat{\mathbf{e}}_{0\neg y}-\hat{\mathbf{e}}_{1\neg y}-\hat{\mathbf{e}}_{0y}+\hat{\mathbf{e}}_{1y}\right) \tag{4.34}$$

$$=\frac{1}{\sqrt{2}}\left(\hat{\mathbf{e}}_{1y}-\hat{\mathbf{e}}_{1\neg y}\right) \tag{4.35}$$

We can simplify this derivation by using the expression for the action of the Hadamard operator. We have for $\mathbf{b}=H_4\mathbf{a}$,

$$\mathbf{b_x}=\frac{1}{2}\sum_{\mathbf{y}}-1^{\mathbf{x\cdot y}}\mathbf{a_y}$$

so if our starting state is $\hat{\mathbf{e}}_{01}$, then we have

$$\begin{aligned}\mathbf{b_x}&=\frac{1}{2}(-1)^{\mathbf{x}\cdot 01}\\&=\frac{1}{2}(-1)^{x\cdot 0\oplus y\cdot 1}\\&=\frac{1}{2}(-1)^{y}\end{aligned}$$

Now we use the definition of our reversible Boolean function F,

$$F=\sum_{xz}(\hat{\mathbf{e}}_x\otimes\hat{\mathbf{e}}_z)(\hat{\mathbf{e}}_x^\dagger\otimes\hat{\mathbf{e}}_{z\oplus f(x)}^\dagger)$$

we have $\mathbf{c}=F\mathbf{b}$,

$$\begin{aligned}\mathbf{c}_{xy}&=\sum_{wz}F_{xy,wz}\mathbf{b}_{wz}\\&=\mathbf{b}_{x(y\oplus f(x))}\\&=\frac{1}{2}(-1)^{y\oplus f(x)}.\end{aligned}$$

Then we finally apply a Hadmard gate only on the first quibit, $\mathbf{d}=(H_2\otimes I)\mathbf{c}$,

$$\begin{aligned}\mathbf{d}_{xy}&=\frac{1}{\sqrt{2}}\sum_{wz}(-1)^{xw}\delta_{yz}\mathbf{c}_{wz}\\&=\frac{1}{2\sqrt{2}}\sum_{w}(-1)^{xw}(-1)^{y\oplus f(w)}.\end{aligned}$$

Now we are again reduced to plugging in values

$$\mathbf{d}_{xy} = \frac{1}{2\sqrt{2}}\left((-1)^{y\oplus f(0)} + (-1)^{x}(-1)^{y\oplus f(1)}\right)$$
$$\mathbf{d}_{xy} = \frac{1}{2\sqrt{2}}(-1)^{y}\left((-1)^{f(0)} + (-1)^{x\oplus f(1)}\right).$$

Now we see that the amplitude for state $\mathbf{d}_{0y}$ is given by

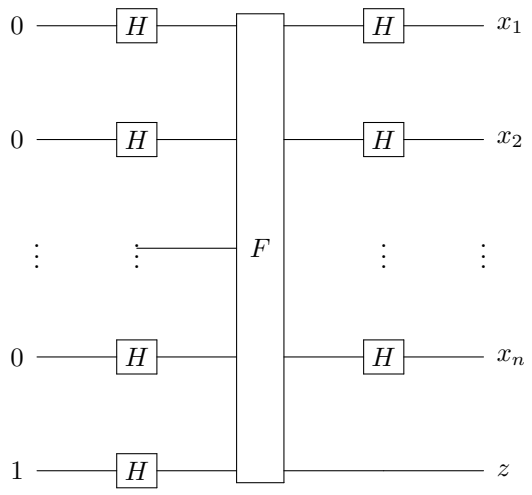
$$\frac{1}{8}\left|-1^{f(0)} + -1^{f(1)}\right|^2$$

so that if f is constant, we have amplitude one half for either state y, whereas if the function changes then the amplitude is 0.

4.1.4 Deutsch-Jozsa Algorithm

Now repeat derivation for n qubits. This is the Deutsch-Jozsa for discriminating between constant and *balanced functions*, which are functions having an equal number of true and false outcomes.

Quantum Circuit:



Functional:

$$(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\,F\,(\mathbf{H}\otimes\cdots\otimes\mathbf{H})\,(\hat{\mathbf{e}}_0\otimes\cdots\otimes\hat{\mathbf{e}}_0\otimes\hat{\mathbf{e}}_1) \tag{4.36}$$

$$=\frac{1}{\sqrt{2N}}\,(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\,F\sum_{(x,z)=0}^{2N-1}e_{xz}\sum_{(x',z')=0}^{2N-1}(-1)^{(x,z)\cdot(x',z')}\delta_{x'0}\delta_{z'1} \tag{4.37}$$

$$=\frac{1}{\sqrt{2N}}\,(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\,F\sum_{(x,z)=0}^{2N-1}e_{xz}(-1)^{(x,z)\cdot(0,1)} \tag{4.38}$$

$$=\frac{1}{\sqrt{2N}}\,(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\,F\sum_{(x,z)=0}^{2N-1}e_{xz}(-1)^{x\cdot 0}(-1)^{z\cdot 1} \tag{4.39}$$

$$=\frac{1}{\sqrt{2N}}\,(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\,F\sum_{(x,z)=0}^{2N-1}e_{xz}(-1)^{z} \tag{4.40}$$

$$=\frac{1}{\sqrt{2N}}\,(\mathbf{H}\cdots\otimes\mathbf{H}\otimes I)\sum_{(x,z)=0}^{2N-1}e_{xz}(-1)^{z\oplus f(x)} \tag{4.41}$$

$$=\frac{1}{\sqrt{2N}}\frac{1}{\sqrt{N}}(\sum_{x'=0}^{N-1}\sum_{x=0}^{N-1}(-1)^{x\cdot x'}e_{x'}e_x)\otimes e_z(-1)^{z\oplus f(x)} \tag{4.42}$$

$$=\frac{1}{\sqrt{2N}}\sum_{(x')=0}^{2N-1}(-1)^{x\cdot x'}(-1)^{z\oplus f(x')}e_{xz} \tag{4.43}$$

Thus, the probability of measuring a state (x,z) is given by

$$P(x,z)=\frac{1}{2N^2}\left|\sum_{x'=0}^{N-1}(-1)^{x\cdot x'}(-1)^{z\oplus f(x')}\right|^2, \tag{4.44}$$

$$=\frac{1}{2N^2}\left|(-1)^z\sum_{x'=0}^{N-1}(-1)^{x\cdot x'}(-1)^{f(x')}\right|^2, \tag{4.45}$$

$$=\frac{1}{2N^2}\left|\sum_{x'=0}^{N-1}(-1)^{x\cdot x'}(-1)^{f(x')}\right|^2. \tag{4.46}$$

If we ask for $P(0,z)$, we get

$$P(0,z)=\frac{1}{2N^2}\left|\sum_{x'=0}^{N-1}(-1)^{0\cdot x'}(-1)^{f(x')}\right|^2, \tag{4.47}$$

$$=\frac{1}{2N^2}\left|\sum_{x'=0}^{N-1}(-1)^{f(x')}\right|^2. \tag{4.48}$$

If f is constant, then we get

$$P(0,z) = \frac{1}{2N^2}\left|\sum_{x'=0}^{N-1} \pm 1\right|^2, \tag{4.49}$$

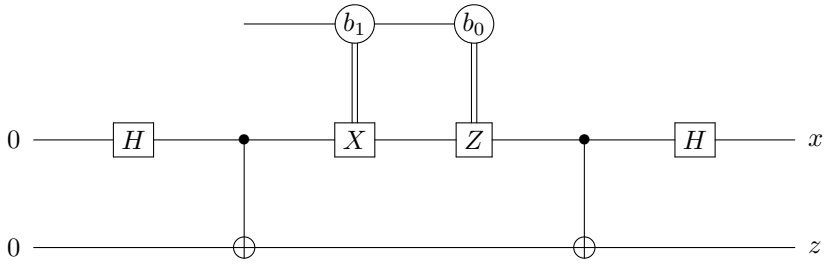
$$= \frac{1}{2N^2}\left|\pm N\right|^2, \tag{4.50}$$

$$= \frac{1}{2}, \tag{4.51}$$

whereas, if f is balanced, then half the terms in the sum cancel the other half, and we get zero.

4.1.5 Superdense coding

Quantum Circuit:



First we work through the case where the U is the identity, so we expect non-zero amplitude only in the (00) state,

$$(\mathbf{H}\otimes I)\mathbf{CNOT}(\mathbf{U}\otimes I)\mathbf{CNOT}(\mathbf{H}\otimes I)\hat{\mathbf{e}}_{00} \tag{4.52}$$

$$=\frac{1}{\sqrt{2}}(\mathbf{H}\otimes I)\mathbf{CNOT}\ \mathbf{CNOT}((\hat{\mathbf{e}}_0+\hat{\mathbf{e}}_1)\otimes\hat{\mathbf{e}}_0) \tag{4.53}$$

$$=\frac{1}{\sqrt{2}}(\mathbf{H}\otimes I)\mathbf{CNOT}\ \mathbf{CNOT}(\hat{\mathbf{e}}_{00}+\hat{\mathbf{e}}_{10}) \tag{4.54}$$

$$=\frac{1}{\sqrt{2}}(\mathbf{H}\otimes I)\mathbf{CNOT}(\hat{\mathbf{e}}_{00}+\hat{\mathbf{e}}_{11}) \tag{4.55}$$

$$=\frac{1}{\sqrt{2}}(\mathbf{H}\otimes I)(\hat{\mathbf{e}}_{00}+\hat{\mathbf{e}}_{10}) \tag{4.56}$$

$$=\frac{1}{\sqrt{2}}(\mathbf{H}\otimes I)((\hat{\mathbf{e}}_0+\hat{\mathbf{e}}_1)\otimes\hat{\mathbf{e}}_0) \tag{4.57}$$

$$=(\hat{\mathbf{e}}_0\otimes\hat{\mathbf{e}}_0) \tag{4.58}$$

$$=\hat{\mathbf{e}}_{00} \tag{4.59}$$

Then $U = X$ which gives (01),

$$\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{X} \otimes I)(\hat{\mathbf{e}}_{00} + \hat{\mathbf{e}}_{11}) \tag{4.60}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\hat{\mathbf{e}}_{10} + \hat{\mathbf{e}}_{01}) \tag{4.61}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)(\hat{\mathbf{e}}_{11} + \hat{\mathbf{e}}_{01}) \tag{4.62}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)((\hat{\mathbf{e}}_0 + \hat{\mathbf{e}}_1) \otimes \hat{\mathbf{e}}_1) \tag{4.63}$$
$$= (\hat{\mathbf{e}}_0 \otimes \hat{\mathbf{e}}_1) \tag{4.64}$$
$$= \hat{\mathbf{e}}_{01} \tag{4.65}$$

$U = Z$ gives (10),

$$\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{Z} \otimes I)(\hat{\mathbf{e}}_{00} + \hat{\mathbf{e}}_{11}) \tag{4.66}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\hat{\mathbf{e}}_{00} - \hat{\mathbf{e}}_{11}) \tag{4.67}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)(\hat{\mathbf{e}}_{00} - \hat{\mathbf{e}}_{10}) \tag{4.68}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)((\hat{\mathbf{e}}_0 - \hat{\mathbf{e}}_1) \otimes \hat{\mathbf{e}}_0) \tag{4.69}$$
$$= (\hat{\mathbf{e}}_1 \otimes \hat{\mathbf{e}}_0) \tag{4.70}$$
$$= \hat{\mathbf{e}}_{10} \tag{4.71}$$

and $U = ZX$ gives (11),

$$\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{ZX} \otimes I)(\hat{\mathbf{e}}_{00} + \hat{\mathbf{e}}_{11}) \tag{4.72}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\hat{\mathbf{e}}_{01} - \hat{\mathbf{e}}_{10}) \tag{4.73}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)(\hat{\mathbf{e}}_{01} - \hat{\mathbf{e}}_{11}) \tag{4.74}$$
$$= \frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)((\hat{\mathbf{e}}_0 - \hat{\mathbf{e}}_1) \otimes \hat{\mathbf{e}}_1) \tag{4.75}$$
$$= (\hat{\mathbf{e}}_1 \otimes \hat{\mathbf{e}}_1) \tag{4.76}$$
$$= \hat{\mathbf{e}}_{11} \tag{4.77}$$

In order to work this out in the general case, we need an expression for U parameterized by the values of the bits $(b_0 b_1)$ that we would like to send across the channel. Let us define the single qubit operator U as

$$U = \begin{pmatrix} (-1)^{b_0 \cdot b_1} \neg b_0 & (-1)^{\neg b_0 \cdot b_1} & b_0 \\ (-1)^{b_0 \cdot b_1} & b_0 & (-1)^{\neg b_0 \cdot b_1} \neg b_0 \end{pmatrix}$$

or equivalently

$$U = (-1)^{b_0 \cdot b_1} \hat{\mathbf{e}}_{b_1} \hat{\mathbf{e}}_0^\dagger + (-1)^{b_0 \cdot \neg b_1} \hat{\mathbf{e}}_{\neg b_1} \hat{\mathbf{e}}_1^\dagger \tag{4.78}$$

Now we can write this out in the general case

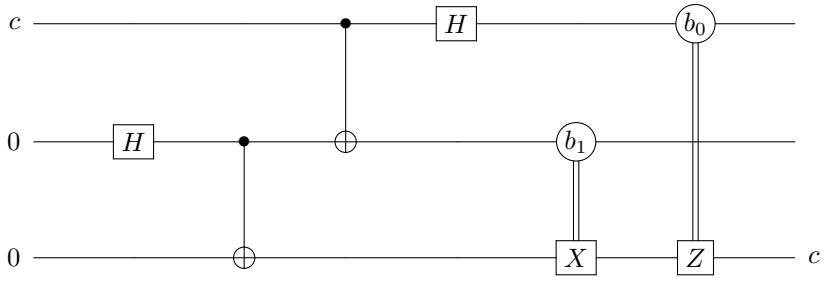
$$\begin{aligned}
&(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{U} \otimes I)\mathbf{CNOT}(\mathbf{H} \otimes I)\hat{\mathbf{e}}_{00} && (4.79)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{U} \otimes I)\mathbf{CNOT}((\hat{\mathbf{e}}_0 + \hat{\mathbf{e}}_1) \otimes \hat{\mathbf{e}}_0) && (4.80)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{U} \otimes I)\mathbf{CNOT}(\hat{\mathbf{e}}_{00} + \hat{\mathbf{e}}_{10}) && (4.81)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}(\mathbf{U} \otimes I)(\hat{\mathbf{e}}_{00} + \hat{\mathbf{e}}_{11}) && (4.82)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)\mathbf{CNOT}((-1)^{b_0 \cdot b_1}\hat{\mathbf{e}}_{b_1 0} + (-1)^{b_0 \cdot \neg b_1}\hat{\mathbf{e}}_{\neg b_1 1}) && (4.83)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)(\neg b_1 \hat{\mathbf{e}}_{00} + b_1(-1)^{b_0}\hat{\mathbf{e}}_{11} + \neg b_1(-1)^{b_0}\hat{\mathbf{e}}_{10} + b_1\hat{\mathbf{e}}_{01}) && (4.84)\\
=&\frac{1}{\sqrt{2}}(\mathbf{H} \otimes I)((\neg b_1 \hat{\mathbf{e}}_0 + \neg b_1(-1)^{b_0}\hat{\mathbf{e}}_1) \otimes \hat{\mathbf{e}}_0 + (b_1(-1)^{b_0}\hat{\mathbf{e}}_1 + b_1\hat{\mathbf{e}}_0) \otimes \hat{\mathbf{e}}_1) && (4.85)\\
=&((\neg b_1(\hat{\mathbf{e}}_0 + \hat{\mathbf{e}}_1) + \neg b_1(-1)^{b_0}(\hat{\mathbf{e}}_0 - \hat{\mathbf{e}}_1)) \otimes \hat{\mathbf{e}}_0\\
&+ (b_1(-1)^{b_0}(\hat{\mathbf{e}}_0 - \hat{\mathbf{e}}_1) + b_1(\hat{\mathbf{e}}_0 + \hat{\mathbf{e}}_1)) \otimes \hat{\mathbf{e}}_1) && (4.86)\\
=&\neg b_1 \frac{1 + (-1)^{b_0}}{2}\hat{\mathbf{e}}_{00} + \neg b_1 \frac{1 - (-1)^{b_0}}{2}\hat{\mathbf{e}}_{10}\\
&+ b_1 \frac{1 + (-1)^{b_0}}{2}\hat{\mathbf{e}}_{01} + b_1 \frac{1 - (-1)^{b_0}}{2}\hat{\mathbf{e}}_{11} && (4.87)
\end{aligned}$$

and check that we obtain the expected amplitudes.

$$\begin{aligned}
(0,0) &\to \hat{\mathbf{e}}_{00}\\
(0,1) &\to \hat{\mathbf{e}}_{01}\\
(1,0) &\to \hat{\mathbf{e}}_{10}\\
(1,1) &\to \hat{\mathbf{e}}_{11}
\end{aligned}$$

4.1.6 Quantum Teleportation

Quantum Circuit:



Let us define the qubit $\mathbf{c} = \alpha\hat{\mathbf{e}}_0 + \beta\hat{\mathbf{e}}_1$, and then write the functional expression for our circuit, evaluating the circuit up to the measurement step,

$$(\mathbf{H}\otimes I\otimes I)(\mathbf{CNOT}\otimes I)(I\otimes\mathbf{CNOT})(I\otimes\mathbf{H}\otimes I)(\mathbf{c}\otimes\hat{\mathbf{e}}_0\otimes\hat{\mathbf{e}}_0) \tag{4.88}$$

$$=(\mathbf{H}\otimes I\otimes I)(\mathbf{CNOT}\otimes I)(I\otimes\mathbf{CNOT})\frac{1}{\sqrt{2}}(\mathbf{c}\otimes(\hat{\mathbf{e}}_0+\hat{\mathbf{e}}_1)\otimes\hat{\mathbf{e}}_0) \tag{4.89}$$

$$=(\mathbf{H}\otimes I\otimes I)(\mathbf{CNOT}\otimes I)\frac{1}{\sqrt{2}}(\mathbf{c}\otimes\hat{\mathbf{e}}_{00}+\mathbf{c}\otimes\hat{\mathbf{e}}_{11}) \tag{4.90}$$

$$=(\mathbf{H}\otimes I\otimes I)(\mathbf{CNOT}\otimes I)\frac{1}{\sqrt{2}}(\alpha\hat{\mathbf{e}}_{000}+\beta\hat{\mathbf{e}}_{100}+\alpha\hat{\mathbf{e}}_{011}+\beta\hat{\mathbf{e}}_{111}) \tag{4.91}$$

$$=(\mathbf{H}\otimes I\otimes I)\frac{1}{\sqrt{2}}(\alpha\hat{\mathbf{e}}_{000}+\beta\hat{\mathbf{e}}_{110}+\alpha\hat{\mathbf{e}}_{011}+\beta\hat{\mathbf{e}}_{101}) \tag{4.92}$$

$$=\frac{1}{2}(\alpha\hat{\mathbf{e}}_{000}+\alpha\hat{\mathbf{e}}_{100}+\beta\hat{\mathbf{e}}_{010}-\beta\hat{\mathbf{e}}_{110}+\alpha\hat{\mathbf{e}}_{011}+\alpha\hat{\mathbf{e}}_{111}+\beta\hat{\mathbf{e}}_{001}-\beta\hat{\mathbf{e}}_{101}) \tag{4.93}$$

$$=\frac{1}{2}\left(\hat{\mathbf{e}}_{00}\otimes(\alpha\hat{\mathbf{e}}_0+\beta\hat{\mathbf{e}}_1)+\hat{\mathbf{e}}_{01}\otimes(\beta\hat{\mathbf{e}}_0+\alpha\hat{\mathbf{e}}_1)\right)$$

$$+\frac{1}{2}\left(\hat{\mathbf{e}}_{10}\otimes(\alpha\hat{\mathbf{e}}_0-\beta\hat{\mathbf{e}}_1)+\hat{\mathbf{e}}_{11}\otimes(-\beta\hat{\mathbf{e}}_0+\alpha\hat{\mathbf{e}}_1)\right) \tag{4.94}$$

Now if we measure the first two qubits, we select one of the states above. We could analyze this case-by-case, but instead let us write a parameterized state based on the bits $(b_0 b_1)$ we get from the measurement

$$\frac{1}{2}\left(\hat{\mathbf{e}}_{b_0 b_1}\otimes(\alpha\hat{\mathbf{e}}_{b_1}+(-1)^{b_0}\beta\hat{\mathbf{e}}_{\neg b_1})\right)$$

and we can act on the last qubit using the operator we defined in Eq. (4.78),

$$\left((-1)^{b_0\cdot b_1}\hat{\mathbf{e}}_{b_1}\hat{\mathbf{e}}_0^\dagger+(-1)^{b_0\cdot\neg b_1}\hat{\mathbf{e}}_{\neg b_1}\hat{\mathbf{e}}_1^\dagger\right)(\alpha\hat{\mathbf{e}}_{b_1}+(-1)^{b_0}\beta\hat{\mathbf{e}}_{\neg b_1}) \tag{4.95}$$

$$=\neg b_1((-1)^{b_0\cdot b_1}\alpha\hat{\mathbf{e}}_{b_1}+(-1)^{b_0\cdot\neg b_1}(-1)^{b_0}\beta\hat{\mathbf{e}}_{\neg b_1})$$

$$+\ \ b_1((-1)^{b_0\cdot b_1}(-1)^{b_0}\beta\hat{\mathbf{e}}_{b_1}+(-1)^{b_0\cdot\neg b_1}\alpha\hat{\mathbf{e}}_{\neg b_1}) \tag{4.96}$$

$$=\neg b_1(\alpha\hat{\mathbf{e}}_0+\beta\hat{\mathbf{e}}_1)+b_1(\beta\hat{\mathbf{e}}_1+\alpha\hat{\mathbf{e}}_0) \tag{4.97}$$

$$=(\neg b_1+b_1)(\alpha\hat{\mathbf{e}}_0+\beta\hat{\mathbf{e}}_1) \tag{4.98}$$

$$=\alpha\hat{\mathbf{e}}_0+\beta\hat{\mathbf{e}}_1 \tag{4.99}$$

$$=\mathbf{c} \tag{4.100}$$

and we have exactly recovered the original state of the first qubit, now in the third qubit.

4.2 Thoughts on Quantum Weirdness

I don't think its the fact that a thing can be in a superposition of states that is weird. It sounds weird if your states are "alive" and "dead", but it sounds perfectly normal if you are in a superposition of red and blue (purple). Our

common experience is filled with objects in a mixed state. The weird thing is that when I measure something, I can only get out one ofthe pure states, red or blue, not the mixed state, purple. So again, the structure of the theory is not weird, its the act of measurement.

Measurement is not really a single act, since quantum systems "measure" themselves all thetime by interacting with other quantum systems. As long as we can fully describe the composite system with quantum mechanics, its not a "measurement". Really, it seems that measurement is the act of bringing a quantum system into equilibrium with another very large quantum system, which destroy correlations it might have had with other systems. I think this is "decoherence". So the question is, how can this kind of equilibration eliminate all states but the pure (basis) states. Suppose the basis states are eigenfunctions of a quantum operator, and the decoherence process involves applying that operator over and over again until we asymptote to fixed point. This kind of power iteration would drive the system to the largest eigenvector represented in the initial state. The analogy is not perfect, since we would need to drive the system to some state depending on the amplitude of that basis vector, but this kind of process could produce the weird behavior of "collapsing" the linear combination to a single basis state.

Interference is not weird, its just a consequence of demanding continuity of any transformation (see Lucien Hardy paper).

4.3 Problems

Problem IV.1 QALA 7.1

Problem IV.2 QALA 7.4

Problem IV.3 QALA 7.5

Problem IV.4 QALA 7.6

Problem IV.5 QALA 8.1

Problem IV.6 QALA 8.2

Problem IV.7 QALA 8.4

Problem IV.8 QALA 9.1

Problem IV.9 QALA 9.2

Problem IV.10 QALA 13.1

Problem IV.11 QALA 13.2

Problem IV.12 QALA 13.3

Problem IV.13 QALA 13.5

Chapter 5

Problem Solutions

Index