Modern Physics: a Modern Approach

Dr. Robert A Ross, University of Detroit Mercy

Robert A. Ross is a Professor of Physics in the Department of Chemistry & Biochemistry at the University of Detroit Mercy. His research interests include semiconductor devices and physics pedagogy. Ross received his B.S. and Ph.D. degrees in Physics from Wayne State University in Detroit.

Modern Physics: A Modern Approach

Abstract

The highest level physics course that an engineer is likely to take is Modern Physics, an upper-level class typically offered by the physics department. This course may be required for electrical or computer engineers as a prelude to a semiconductor device class.

Surveys of textbooks indicate that the curriculum for such courses has not changed much in a couple of decades. The table of contents of a typical text includes such topics as: relativity, quantization, the Bohr-Rutherford nuclear atom, wave-particle duality and the Heisenberg uncertainty principle, the Schrödinger equation, atomic physics, statistical physics, and various related applications. Virtually none of the popular texts include topics on quantum entanglement and quantum computing.

Quantum entanglement involves correlations in the measurement of physical systems. These systems are often electrons, photons, or nuclei but even mechanical oscillators can exhibit entanglement.[1] In any case, entanglement is the key feature of quantum computing systems and quantum cryptographic information transmission. As such, it is a topic of active current research and development.

In our paper we will describe a new method to introduce quantum mechanics, prior to the introduction of the Schrödinger equation. We use the results of recent published research in physics education to introduce students to the concept of simulated quantum computation. No claim is made toward the development of a quantum computer; instead we use a novel technique to simulate its operation. Students are introduced to the concept of a quantum bit or qubit and how the qubits are represented mathematically and how various operators affect them. They use linear algebra and matrix techniques that are familiar to students to develop a novel understanding of that which is quite unfamiliar – quantum mechanics.

Introduction

The College of Engineering & Science at the University of Detroit Mercy offers a junior level physics course. Modern Physics with Device Applications is required for electrical engineers and is an elective for other engineering students. The calculus-based general physics sequence is the prerequisite requirement of the course. All engineers are required to take Engineering Computing and Problem Solving in their freshman year. This course introduces them to the fundamentals of MATLAB so students are well prepared for its application in the modern physics class.[2]

Last year at the ASEE annual conference Choi argued that the current modern physics courses are woefully inadequate, "typical quantum physics courses we offer to engineering students are not even adequate for the *existing* technologies that have brought us the Information Age, much less in preparing our students for another big transformation we are currently witnessing."[3] We agree wholeheartedly and believe that by introducing elements of quantum computation into the curriculum we can motivate students to learn something about quantum mechanics and provide enough background so that they have some ability to read the current literature. The ability to read any article about quantum computation or quantum mechanics requires some knowledge of Dirac notation. Coverage of this is sorely lacking in current modern physics textbooks. We have revised our modern physics course to include some quantum computing projects. These introductory projects can easily be integrated into the curriculum since our engineering students have familiarity with MATLAB. The MATLAB environment is designed to handle vectors and matrices – the fundamental tools of quantum mechanics. Quantum mechanics is introduced by describing entanglement from the very beginning. Mermin's device is used as a platform to delve into the extremely peculiar nature of quantum mechanics.[4]

This paper is organized as follows: first we describe Mermin's device, next we explain how Dirac notation is introduced into the curriculum and subsequently we present four student projects developed from recently published work and conclude with some samples of code written by students. [5]

Mermin's Quantum Device

To reacquaint, or introduce, the reader to Mermin's thought experiment we present a description of the results of its operation. The reader is encouraged to read the original paper in reference 2. The device consists of a transmitter and two receivers. Call the operators of the receivers Alice and Bob. The transmitter and receivers have no connections with each other so no communication is possible between them. When the button at the top of the transmitter is pressed, two particles or two waves or two whatever are emitted toward Alice and Bob. The receivers have a switch with three possible settings 1, 2, and 3 and two lights. One light is red and the other green. The transmitter and receivers are shown below in Fig. 1.

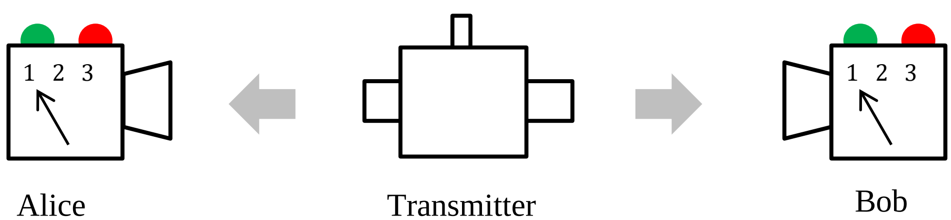


Fig. 1
Schematic representation of David Mermin's quantum device with a transmitter and two receivers operated by Alice and Bob as shown.

Alice and Bob each set the switches on their respective receivers to position 1, 2 or 3 at random. After the switches are set, the button on the transmitter is depressed and *"things"* travel from the transmitter to the receivers causing one of the lights on each receiver to flash red or green. Since the switch positions are set at random all pairs of settings are equally likely: 11, 12, 13, 21, 22, 23, 31, 32, and 33. If Alice's detector was set to 1 and flashed green she records 1G. If Bob's detector was set to 2 and flashed red he records the data as 2R. What are the results of numerous runs of the device? Alice and Bob both observe that the red light flashes as frequently as the green light and there is no pattern or correlation between the switch settings and the light flashes. The data that is collected by Alice and Bob are completely random. They each observe 1R, 2R, 3R, 1G, 2G, and 3G with equal frequency. Of course since the switch settings are random the data has a random character like tossing a coin. It is possible for 1R to occur several times in a row just like it is possible to toss a coin and have heads appear three times in a row. After a large number of runs, the probability of each outcome 1R, 3G, etc. is the same.

The fun begins when Alice and Bob combine their data for each run after the device has been operated many times. When they combine their data they put it in the form: 11RR, 12GG, 23RG, etc. The result 13GR means Alice's switch setting was 1 and her light flashed green while Bob's switch setting was 3 and his light flashed red. The data from 100 runs of Mermin's device is shown below in Table I.

31GR	12GG	21GR	33RR	22GG	33RR	23RR	22RR	11GG	22RR
23GR	13GR	11GG	31GR	22GG	22GG	11GG	31GG	22RR	23GG
31GG	23RG	12RR	12GR	13RG	13GR	13GR	31RG	12RG	21RG
21GR	12GR	31RG	33RR	23RG	31GR	33RR	11RR	33GG	22RR
23RG	13GR	32GG	32GR	32GR	12GG	23RR	33GG	12GR	31RG
32RG	31GR	13GR	22GG	31GR	32GR	23GR	21GR	32RG	32GR
33RR	12RG	31RG	13RG	12RR	33RR	33RR	11GG	11GG	22GG
13GR	33GG	22RR	13RG	12GR	32RG	31GR	31RR	33GG	33RR
32GR	21GR	23RG	32RR	32GR	33GG	22RR	13GR	31RG	11GG
22RR	31GG	32GR	31RR	32RR	31GG	12RR	12GR	32RG	23RG

Table I
Simulated results from 100 runs of Mermin's device with the data recorded in the format: Alice's switch setting; Bob's switch setting; Alice's color reading; Bob's color reading.

An analysis of the data indicates that we can group the results into two categories: one where the switches have the same settings and one where the switch settings are different.

Case a) In those runs where the switches have the same setting (11, 22, or 33) Alice and Bob observe that the lights flash the same color: RR and GG with equal frequency but RG and GR never occur.

Case b) In those runs in which the switches end up with different settings (12, 13, 21, 23, 31, or 32) Alice and Bob observe their detectors flash the same color only a quarter of the time (RR and GG occurring with equal frequency); the other three quarters of the time the detectors flash different colors (RG and GR occurring with equal frequency).

How can these results be explained? Mermin cleverly argues that the only way to explain the results is to infer that whatever is travelling from the transmitter to the receiver must contain "an instruction set" based on some physical property (or element of physical reality)[6] to tell the receiver which light to flash for a given switch position. When Alice's switch and Bob's switch are in the same position the same color light flashes can be explained by hypothesizing that two things moving between the transmitter and Alice's detector and Bob's detector have identical instruction sets. There are eight instruction sets: RRR, RRG, RGR, RGG, GRR, GRG, GGR, and

GGG. If the instruction set is RGR when the switch setting is 1 it flashes red when it is 2 it flashes green and when it is 3 it flashes red. The instruction RRR will cause the red light to flash for all three possible switch settings. There is a problem. When the switches have different settings RR and GG occur only a quarter of the time. The only instruction sets where the lights can flash different colors are RRG, RGR, GRR, GGR, GRG, or RGG. Suppose the instruction is RRG then for Alice's switch setting 1 and Bob's switch setting 2 or Alice's switch setting 2 and Bob's switch setting 1 both Alice and Bob will observer their lights flashing red. For the other 4 possible cases Alice = 1 and Bob = 3, or Alice = 3 and Bob = 1 or , Alice = 2 and Bob = 3, or Alice = 3 and Bob = 2 both Alice and Bob will see their light flashing different colors. Therefore if the instruction set is RRG then Alice and Bob will see their lights flashing the same color 2 out of 6 times or 1/3 of the time. The same argument is true for the other 5 instruction sets: RGR, RGG, GRR, GRG, and GGR. We are forced to conclude that instruction sets require that in cases where the switch settings are different the lights will flash the same color at least one-third of the time–it will be more than one-third if the instruction sets RRR or GGG ever occur. Note that Mermin subsequently "revisits" and "refines" this discussion with thought experiments that do not require perfect correlations between the particles.[7,8] In our modern language we would say that the "*things*" that traverse the path from transmitter to receiver have undergone "*quantum entanglement.*" Schroeder points out that the word entanglement was first introduced by Schrödinger in 1935 but has been virtually absent from publication until the 1980's.[9]

Dirac Notation – Vectors and Operators/Matrices

Paul Dirac introduced a new mathematical notation to describe quantum mechanics that has proven to be durable and robust.[10,11] Anyone that chooses to read the current literature needs some familiarity with the language; however this is not a major obstacle since students are often familiar with some concepts from the study of vectors and/or linear algebra. The following material is what is presented in class.

Kets, $|\ \rangle$, are represented by column vectors such as $|A\rangle = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix}$, in principle they could have an infinite number of elements (we needn't worry about that detail) and the A_i are complex numbers. In a similar fashion *bras*, $\langle\ |$, are row vectors $\langle B| = \begin{pmatrix} B_1 & B_2 & B_3 \end{pmatrix}$ which also could have an infinite number of elements. A *bra–ket* is like a dot or inner product $\langle B|A\rangle = \begin{pmatrix} B_1 & B_2 & B_3 \end{pmatrix}\begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} = B_1A_1 + B_2A_2 + B_3A_3$. One can turn a ket into a bra by converting it to a row vector and taking the complex conjugate, which is represented by the superscript *, so $\langle A|A\rangle = \begin{pmatrix} A_1^* & A_2^* & A_3^* \end{pmatrix}\begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} = A_1^*A_1 + A_2^*A_2 + A_3^*A_3$ ends up being real and positive.

Linear operators are used to transform, or operate on, bras and kets.

The identity matrix, $\hat{I}$, has ones along the diagonal and zeros everywhere else, it is represented in 2×2 form as $\hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. All quantum mechanical operators are represented by unitary matrices; note that we put a caret on the matrix for identification, e.g. $\hat{U}$. The superscript "dagger" † symbol represents the adjoint – the complex conjugate transpose. (The transpose interchanges the rows and columns.) A matrix is unitary if $\hat{U}\hat{U}^\dagger = \hat{U}^\dagger\hat{U} = \hat{I}$. It follows that $\left(\hat{A}+\hat{B}\right)^\dagger = \hat{A}^\dagger + \hat{B}^\dagger$ and $\left(\hat{A}\hat{B}\right)^\dagger = \hat{B}^\dagger\hat{A}^\dagger$ and $\left(\hat{A}^\dagger\right)^{-1} = \left(\hat{A}^{-1}\right)^\dagger$. If $\hat{A}^\dagger = \hat{A}$ then $\hat{A}$ is called "Hermitian." All operators in quantum mechanics are Hermitian, they have real eigenvalues.

We operate on a ket from the left. Operating on a ket yields another ket:

$$\hat{U}\left|B\right\rangle = \begin{pmatrix} \hat{U}_{11} & \hat{U}_{12} & \hat{U}_{13} \\ \hat{U}_{21} & \hat{U}_{22} & \hat{U}_{23} \\ \hat{U}_{31} & \hat{U}_{32} & \hat{U}_{33} \end{pmatrix}\begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} \hat{U}_{11}B_1 + \hat{U}_{12}B_2 + \hat{U}_{13}B_3 \\ \hat{U}_{21}B_1 + \hat{U}_{22}B_2 + \hat{U}_{23}B_3 \\ \hat{U}_{31}B_1 + \hat{U}_{32}B_2 + \hat{U}_{33}B_3 \end{pmatrix}$$

We operate on a bra from the right. Operating on a bra yields another bra:

$$\left\langle A\right|\hat{U} = \begin{pmatrix} A_1 & A_2 & A_3 \end{pmatrix}\begin{pmatrix} \hat{U}_{11} & \hat{U}_{12} & \hat{U}_{13} \\ \hat{U}_{21} & \hat{U}_{22} & \hat{U}_{23} \\ \hat{U}_{31} & \hat{U}_{32} & \hat{U}_{33} \end{pmatrix}$$

$$\left\langle A\right|\hat{U} = \left(A_1\hat{U}_{11} + A_1\hat{U}_{12} + A_1\hat{U}_{13} \quad A_2\hat{U}_{21} + A_2\hat{U}_{22} + A_2\hat{U}_{23} \quad A_3\hat{U}_{31} + A_3\hat{U}_{32} + A_3\hat{U}_{33}\right).$$

We can form the product of a ket and a bra. It will be an operator

$$\left|A\right\rangle\left\langle B\right| = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix}\begin{pmatrix} B_1 & B_2 & B_3 \end{pmatrix} = \begin{pmatrix} A_1B_1 & A_1B_2 & A_1B_3 \\ A_2B_1 & A_2B_2 & A_2B_3 \\ A_3B_1 & A_3B_2 & A_3B_3 \end{pmatrix}$$

These kets and bras are linear $(a+b)\left|\psi\right\rangle = a\left|\psi\right\rangle + b\left|\psi\right\rangle$, this is an important property.

Qubits

A qubit is stored by a quantum computer with 2 basis vectors in "Hilbert space." We are considering the *z*-component of an electrons magnetic moment which is always $S_z = \pm\frac{\hbar}{2}$ which we represent as $\left|\uparrow\right\rangle$ for "spin up" and $\left|\downarrow\right\rangle$ for "spin down". Instead of representing spins, we could represent photons with horizontal or vertical polarization states. $\left|H\right\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\left|V\right\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

We represent the two states of one qubit as $|\uparrow\rangle = |0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$, and $|\downarrow\rangle = |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$.

We notice that:

$$\langle\uparrow|\downarrow\rangle = \langle 0|1\rangle = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = 0, \qquad \langle\downarrow|\uparrow\rangle = \langle 1|0\rangle = \begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = 0,$$

$$\langle\uparrow|\uparrow\rangle = \langle 1|1\rangle = \begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = 1 \qquad \langle\downarrow|\downarrow\rangle = \langle 0|0\rangle = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = 1.$$

These states are orthonormal. The same thing would happen with the horizontal or vertical polarization states $\langle H|H\rangle = \langle V|V\rangle = 1$ and $\langle H|V\rangle = \langle V|H\rangle = 0$.

The general quantum state of a spin ½ particle (or photon) is a linear superposition such as $|\psi\rangle = a|\uparrow\rangle + b|\downarrow\rangle = a|1\rangle + b|0\rangle$ where *a* and *b* are complex amplitudes that obey the Born[12] rule, $|a|^2 + |b|^2 = a^* a + b^* b = 1$. They are a superposition of states – something with no classical analog. An *N*-qubit register is *N* of these 2-state systems. A 3-qubit register has 2^3 basis states. For an *N*-qubit register with 2^N basis states a quantum operator (quantum logic gate) is represented by a $2^N \times 2^N$ unitary matrix.

For a 2-state system that corresponds to 2^N basis states; we consider a 3-qubit register that has 2^3= 8 basis states. The basis states are formed by taking the tensor product, $\otimes$, of the individual basis states. MATLAB makes this operation simple with the *kron* function.

Tensor Product $\otimes$

With 2-component column vectors *x* and *y* we can write the tensor product as:

$$x \otimes y = \begin{pmatrix}x_0\\x_1\end{pmatrix} \otimes \begin{pmatrix}y_0\\y_1\end{pmatrix} = \begin{pmatrix} x_0\begin{pmatrix}y_0\\y_1\end{pmatrix} \\ x_1\begin{pmatrix}y_0\\y_1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}x_0 y_0\\x_0 y_1\\x_1 y_0\\x_1 y_1\end{pmatrix}$$

which is a ket, a column vector. The tensor product multiplies every element on the left with the quantity on the right. It works for matrices too.

$a = \begin{pmatrix}a_{11} & a_{12}\\a_{21} & a_{22}\end{pmatrix}$, $b = \begin{pmatrix}b_{11} & b_{12}\\b_{21} & b_{22}\end{pmatrix}$. We insert the matrix for the quantity on the right next to each element of the matrix on the left.

$$a \otimes b = \begin{pmatrix}a_{11}b & a_{12}b\\a_{21}b & a_{22}b\end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12}\\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22}\\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{11}\\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{11} & a_{22}b_{11}\end{pmatrix}.$$

Very often the tensor product symbol, $\otimes$, is omitted.

3–Qubit systems

This system is made up of tensor products of the form:
$|0\rangle \otimes |0\rangle \otimes |0\rangle$, $|0\rangle \otimes |0\rangle \otimes |1\rangle$, **...**, and $|1\rangle \otimes |1\rangle \otimes |1\rangle$**.** For clarity we show the first state as,

$$|0\rangle \otimes |0\rangle \otimes |0\rangle = |0\rangle|0\rangle|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1\begin{pmatrix}1\\0\end{pmatrix}\\0\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\\0\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} = |000\rangle$$

Notice that these kets have zeros everywhere except for one place. If we index the array, or vector, from 0 to 7 then the 1 appears in the location given by the decimal value of the state in this representation. For example, in binary 000 = 0, 001 = 1, 010 = 2, etc. The general quantum state is a superposition of all possible states and is given by,
$|\psi\rangle = a|000\rangle + b|001\rangle + ... + g|110\rangle + h|110\rangle$, where *a, b, c,* ... are complex amplitudes and $|a|^2 + |b|^2 + ... + |g|^2 + |h|^2 = 1$. We store $|\psi\rangle$ as a column vector with 8 complex amplitudes:

$$|\psi\rangle = \begin{pmatrix}a\\b\\c\\d\\e\\f\\g\\h\end{pmatrix}.$$

Simulated Measurement of 3-qubit register: measuring $|\psi\rangle$.

Candela (see reference 4) shows how to simulate the measurement of a general quantum state – in this case the 3 qubit simulated quantum register $|\psi\rangle$. A graphical representation of the measurement of a 3-qubit quantum register is shown below in Fig. 2. The 3 qubits are represented by the horizontal lines, the boxes represent logic gates which are $2^3 \times 2^3 = 8\times 8$ unitary matrices representing an operator. The qubit register is initialized to the state $|\psi\rangle = |000\rangle$. No operations are performed on qubit one. On qubit two, the unitary matrix $\hat{U}_1$ operates on $|\psi\rangle$

first followed by $\hat{U}_2$. The matrix $\hat{U}_3$ operates on the third qubit and subsequently the new resulting state is measured. The little voltmeters represent the quantum measurement. The operators $\hat{U}_i$ are arbitrary in this case and are used only to help understand the diagram. The result of the measurement is always a basis state of $|\psi\rangle$, i.e. it is one of the kets $|000\rangle$, $|001\rangle$, etc. The algorithm is quite simple and easy to implement in MATLAB. The instructions given to the students are shown later in the paper and are listed as projects.

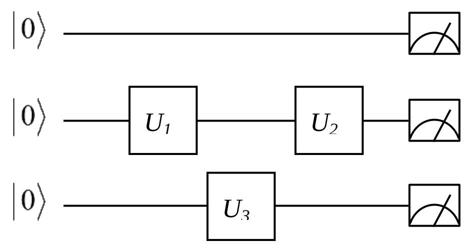


Fig. 2
Schematic representation of a series of operations on a 3-qubit quantum register. No operations are performed on qubit one while qubit two is first operated on by $\hat{U}_1$ followed by $\hat{U}_2$. The matrix $\hat{U}_3$ operates on the third qubit; the state is subsequently measured.

Useful operators: Hadamard and phase-shift

The Hadamard gate, or operator, $\hat{H}$ acts on a single qubit, it is given by $\hat{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. On the schematic diagram it is shown as, $\boxed{H}$ when it operates on a basis state it yields:

$\hat{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ an equal superposition of the two basis states. In a similar fashion it is easy to show that $\hat{H}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$. Suppose that $|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ then a brief calculation shows that $\hat{H}|\psi\rangle = |0\rangle$ which means $\hat{H}\hat{H}|0\rangle = |0\rangle$ and $\hat{H}\hat{H}|1\rangle = |1\rangle$. The Hadamard operator puts a basis state into a superposition and it also puts the superposition back to the original basis state. This is very strange. The Hadamard operator puts a basis state into an equal superposition of $|0\rangle$ and $|1\rangle$. That means if we were to repeatedly measure this new state we would get the result $|0\rangle$ or $|1\rangle$ completely at random and with equal probability. The Hadamard operator introduces randomness into the new state. How can operating on it again, with the same operator, put it back? The second application "undoes" the randomness introduced by the first application of the operator. If we want to apply this gate to the different qubits we need to generate the appropriate operators. The tensor operations needed to form the gate are shown in Table II.

The phase-shift gate, or operator, $\hat{R}_\theta$ acts on a single qubit and is given by $\hat{R}_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$, on the schematic diagram it is shown as $\boxed{\pi}$ where the number in the box represents the phase-shift angle, here π radians. If $|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$, then $\hat{R}_\theta |\psi\rangle = \begin{pmatrix} a \\ be^{i\theta} \end{pmatrix}$. The tensor multiplication operations needed to form this gate are also shown in Table II. The 8×8 representation of all of these gates are given in Appendix B.

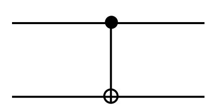
	operator	construction	operator	construction
Qubit 1	$\hat{H}^{(1)}$	$\hat{H}\otimes\hat{I}\otimes\hat{I}$	$\hat{R}_\theta^{(1)}$	$\hat{R}_\theta\otimes\hat{I}\otimes\hat{I}$
Qubit 2	$\hat{H}^{(2)}$	$\hat{I}\otimes\hat{H}\otimes\hat{I}$	$\hat{R}_\theta^{(2)}$	$\hat{I}\otimes\hat{R}_\theta\otimes\hat{I}$
Qubit 3	$\hat{H}^{(3)}$	$\hat{I}\otimes\hat{I}\otimes\hat{H}$	$\hat{R}_\theta^{(3)}$	$\hat{I}\otimes\hat{I}\otimes\hat{R}_\theta$

Table II
Tensor operations required to form a Hadamard operator or phase-shift operator operating on a specific qubit. The qubit is denoted by the superscript in parenthesis.

Another useful operator: 2-qubit controlled NOT gate or CNOT

This gate acts like a classical exclusive OR gate, XOR, except one qubit controls what happens to the other qubit. Suppose we work in the 2-qubit basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ and we identify an operator that uses one qubit to control what happens to the other. Here the qubit 1, the control qubit, controls qubit 2, the target qubit. The NOT function changes a $|0\rangle$ to a $|1\rangle$ and a $|1\rangle$ to a $|0\rangle$.

This is the symbol for a CNOT gate where the first qubit is the controlling qubit, $\hat{C}_{NOT}^{(1,2)}$.



It operates according to: $\hat{C}_{NOT}^{(1,2)}|00\rangle = |00\rangle$, $\hat{C}_{NOT}^{(1,2)}|01\rangle = |01\rangle$, $\hat{C}_{NOT}^{(1,2)}|10\rangle = |11\rangle$, and $\hat{C}_{NOT}^{(1,2)}|11\rangle = |10\rangle$. The 2-qubit representation is given by:

$$\hat{C}_{NOT}^{(1,2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

If the first qubit is $|0\rangle$ then the second qubit is unchanged. If the first qubit is $|1\rangle$ then the second is flipped.

We can also use the second qubit as the control, a $\hat{C}_{NOT}^{(2,1)}$ gate will function according to: $\hat{C}_{NOT}^{(2,1)}|00\rangle=|00\rangle$, $\hat{C}_{NOT}^{(2,1)}|01\rangle=|11\rangle$, $\hat{C}_{NOT}^{(2,1)}|10\rangle=|10\rangle$, and $\hat{C}_{NOT}^{(2,1)}|11\rangle=|01\rangle$. The 2-qubit representation is given by:

$$\hat{C}_{NOT}^{(2,1)}=\begin{pmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{pmatrix}.$$

If we have N = 3 qubits, any of the three qubits can be the controlling qubit and either of the remaining two can be the controlled qubit, so there are six possible CNOT gates. Here is the matrix for qubit 2 controlling qubit 3 and the matrix for qubit 2 controlling qubit 1.

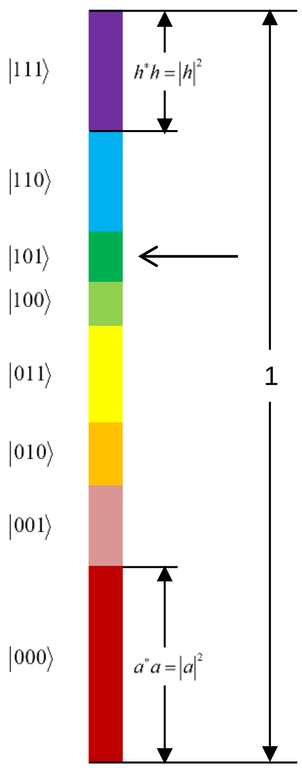
$$\hat{C}_{NOT}^{(2,3)}=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}\qquad \hat{C}_{NOT}^{(2,1)}=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\end{pmatrix}.$$

The $\hat{C}_{NOT}^{(2,3)}$ gate uses qubit 2 to control qubit 3, the target. If qubit 2, the control is $|0\rangle$ it leaves qubit 3 alone, if it is in state $|1\rangle$ it flips the state of qubit 3. It essentially does the following: $\hat{C}_{NOT}^{(2,3)}|000\rangle=|000\rangle$, $\hat{C}_{NOT}^{(2,3)}|001\rangle=|001\rangle$, $\hat{C}_{NOT}^{(2,3)}|010\rangle=|011\rangle$ $\hat{C}_{NOT}^{(2,3)}|011\rangle=|010\rangle$. If qubit 2 is operated on with the Hadamard operator and it was initially in state $|0\rangle$; $\hat{H}|0\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$ and if qubit 3 is $|0\rangle$ the result is $\frac{1}{\sqrt{2}}\left(|000\rangle+|011\rangle\right)$; qubit 2 is entangled with qubit 3.

It is important to recognize that the programming projects listed below are taken directly from reference 4. There is no claim that they were developed by the current author.

Project 1: Initial quantum simulation

Create a MATLAB subroutine or function (probably better) the performs a measurement of the ket $|\psi\rangle$. You pass the routine $|\psi\rangle$ and it returns an integer that represents the state.



The measurement process is equivalent to having a bar divided into eight segments—shown here as different colors. The total length of the bar is exactly 1, the length of the lowest segment (deep red) is $a^*a = |a|^2$ and this corresponds to state $|000\rangle$. The length of the top purple segment is $h^*h = |h|^2$ and it corresponds to state $|111\rangle$. The measurement process is equivalent to dropping an arrow at random and seeing in to which colored bin it falls. This particular measurement produced the state $|101\rangle$.

1. Pick a random number, *R*, such that $0 \le R \le 1$.
2. Keep *R* fixed and do the following:

   Set $q = a^*a = |a|^2$. If $R < q$ then $|\psi\rangle = |000\rangle$, else

   set $q = q + |b|^2$. If $R < q$ then $|\psi\rangle = |001\rangle$, else

   set $q = q + |c|^2$. If $R < q$ then $|\psi\rangle = |010\rangle$, else

   set $q = q + |d|^2$. If $R < q$ then $|\psi\rangle = |011\rangle$, else

   set $q = q + |e|^2$. If $R < q$ then $|\psi\rangle = |100\rangle$, else

   set $q = q + |f|^2$. If $R < q$ then $|\psi\rangle = |101\rangle$, else

   set $q = q + |g|^2$. If $R < q$ then $|\psi\rangle = |110\rangle$, else

   $|\psi\rangle = |111\rangle$.

(Of course $q = q + |b|^2$ means replace q with $q + |b|^2$, you are incrementing q.)
You could write a subroutine or create a function where you pass it an array, $|\psi\rangle$, and it returns a state such as $|100\rangle$. Note that MATLAB indexes arrays beginning with 1 so if you measure state $|000\rangle$ you could return the number 1 (index 1 of the array), if you measure state $|001\rangle$ you could return the number 2. This function will be embedded in code where you will need to keep track of how many times a given state is returned.

1. Allocate a column vector with $2^N = 8$ complex entries call it $|\psi\rangle$ (whatever array name you choose.). Initialize the state $|\psi\rangle$ to a particular value, we'll see in a bit.
2. Generate a simulated result of a measurement (discussed above) of S_z for the three qubits. The results are random, like tossing a coin, but must always yield one of the basis states $|000\rangle, |001\rangle, |010\rangle, \ldots, |111\rangle$. The results should follow the quantum mechanical Born (after Max Born) normalization rule $|a|^2 + |b|^2 + \ldots + |g|^2 + |h|^2 = 1$. If result is $|010\rangle$, this means the first and third qubit are 0 (spin-up) and the second is 1 (spin-down.) You should show the data in a tabular form in the manner of your choice. There is one possibility shown below.

Repeat the measurement–step 2 above–many times to see how the results vary. Each measurement is like a roll of the dice. You need quite a few attempts to be sure the dice are fair, i.e. the probability of any number occurring is 1/6. The same thing is happening here; you measure $|\psi\rangle$ and the result is one of the basis states, you need lots of measurements to see the probability of any particular state occurring. You have a fast computer so you should be able to measure 100,000 times to generate some good statistics.

State	Number of occurrences	Percent of occurrences
$\|000\rangle$	10,000	10
$\|001\rangle$	5,000	5
$\|010\rangle$	25,000	25
$\vdots$	$\vdots$	$\vdots$
$\|111\rangle$	17,000	17
Total	100,000	

To make sure you have a properly working program try the following:

a) Initialize $|\psi\rangle$ to one of the basis states, say $|101\rangle$. The result of measurement should always be $|101\rangle$.

b) Initialize $|\psi\rangle$ to another one of the basis states (not $|101\rangle$ this time). The result of measurement should always be the state you used to initialize $|\psi\rangle$.

c) Initialize $|\psi\rangle$ to the "cat" state–Schrödinger's Cat. You should measure $|000\rangle$ and $|111\rangle$ with equal frequency. $|\psi_{CAT}\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}$

d) Initialize $|\psi\rangle$ to an equal superposition of all 2^N states, $|\psi\rangle = \frac{1}{\sqrt{8}}\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix}$. You should measure all states with equal frequency.

Project 2: Quantum computations

- Write code to run the four quantum computations shown below (in Fig.3).
- **Explain what you expect to measure.** You should have a pretty good idea what happens to each qubit from the first homework assignment.
- Perform the measurements and reconcile with your expectations. Provide the data in a neat tabular form.
- You can also include the .m files to show that this works. Those files should be well documented so it is clear what they are supposed to do.

Gates are applied to qubits by successively operating on $|\psi\rangle$ from the left—the way an operator operates on a ket. Initialize $|\psi\rangle$ to the state $|000\rangle$. Naturally this does not mean all the elements of the ket, $|\psi\rangle$, are zero. The top line is for operations on qubit 1, second line for operations on qubit 2, etc. Reading along the line indicates how the quantum gates, operators, are applied to the initial ket. The little meter represents a quantum measurement.

In each case your code should do something like:

Generate appropriate quantum operators
Check that they are Hermitian

For i=1 to NumberLoop

Initialize the ket $|\psi\rangle$

Operate on $|\psi\rangle$ with the appropriate operators

Measure $|\psi\rangle$

Keep a tally of the result of measurement

endloop

Display results in an exciting manner.

(a) The operation is $\hat{H}^{(2)}|\psi\rangle$. It puts qubit 2 in an equal superposition of $|0\rangle$ and $|1\rangle$. The result of measurement should be $|000\rangle$ and $|010\rangle$ occurring with equal frequency.

(b) The operation is $\hat{H}^{(3)}\hat{H}^{(2)}\hat{H}^{(1)}|\psi\rangle$. Hadamard gates are applied to qubits 1, 2, and 3. Each qubit is equally likely to be in state $|0\rangle$ or $|1\rangle$ with no correlations between any of the qubits. Knowing the value of qubit 1 tells you nothing about qubit 2 and so on. The result should vary randomly between all eight possibilities, $|000\rangle,|001\rangle,|010\rangle,\ldots,|111\rangle$. Putting the *N*-qubit register into an equal superposition of all 2^N basis states by applying a Hadamard gate to each qubit is an important building block of many quantum algorithms.

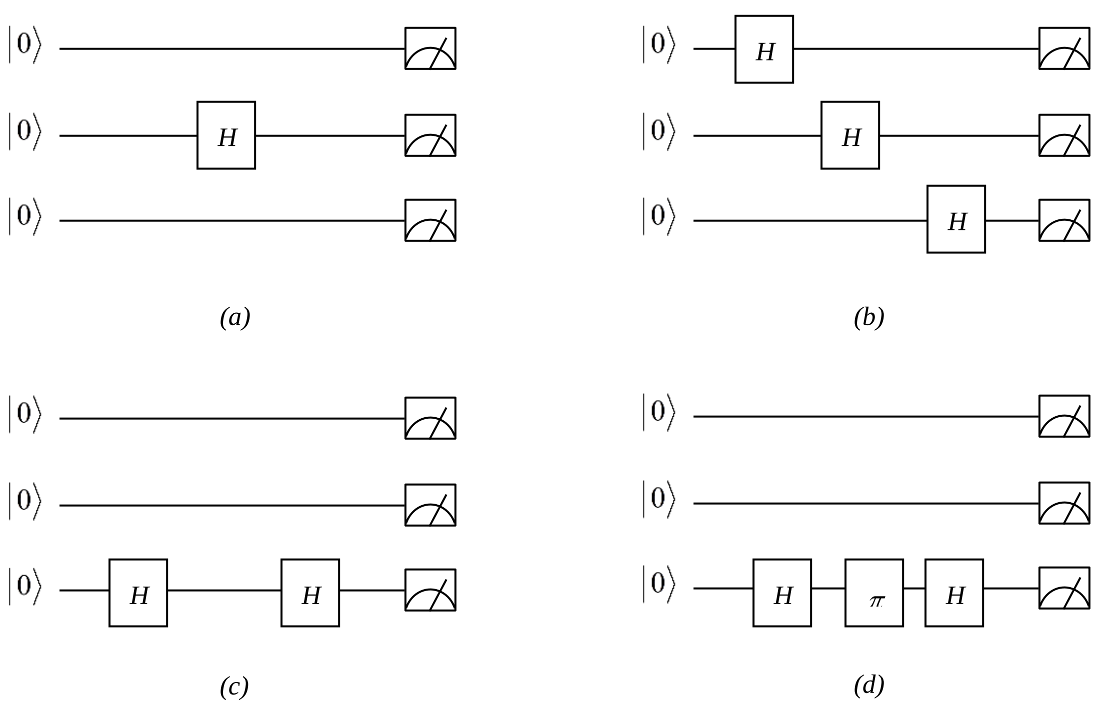


Fig. 3
Schematic representation of four quantum computations.

(c) The operation is $\hat{H}^{(3)}\hat{H}^{(3)}|\psi\rangle$. Two successive Hadamard gates applied to the same qubit. The gate splits a single state into two, but it also puts two states back together into one. The result of the calculation should always be $|000\rangle$.

(d) The operation is $\hat{H}^{(3)}\hat{R}_{\pi}^{(3)}\hat{H}^{(3)}|\psi\rangle$. Two Hadamard gates are applied to a qubit but between them is a phase-shift gate with $\theta = \pi$. The first Hadamard gate puts the $|0\rangle$ into a superposition of $|0\rangle$ and $|1\rangle$. The phase-shift gate changes the phase of the superposition and the second Hadamard gate puts into the state $|1\rangle$. The result of measurement should always be $|001\rangle$. $\hat{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\hat{R}_{\pi}\hat{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \hat{H}|1\rangle$, and so $\hat{H}\hat{R}_{\pi}\hat{H}|0\rangle = \hat{H}\hat{H}|1\rangle = |1\rangle$

Project 3: Grover's Quantum Search Algorithm

Figure 4 shows the quantum circuit for Grover's search algorithm. The size of the database that can be searched is $D = 2^N$, where N is number of qubits. The figure is for $N = 3$ so that $D = 8$. In addition to the Hadamard gates an operator $\hat{O}$ for the quantum oracle and a special operator $\hat{J}$ are needed. The oracle knows what the right question is to give a *yes* or *true* response, and it functions as follows:

- If the oracle is given one of the D-1 wrong questions, it returns the input state unchanged. For example say the right question (represented by a binary number) is 110. Then, since 100 is not the right question, $\hat{O}|100\rangle = |100\rangle$.
- If the oracle is given the right question, it returns the input multiplied by –1. If the right answer is 110 then $\hat{O}|110\rangle = -|110\rangle$.

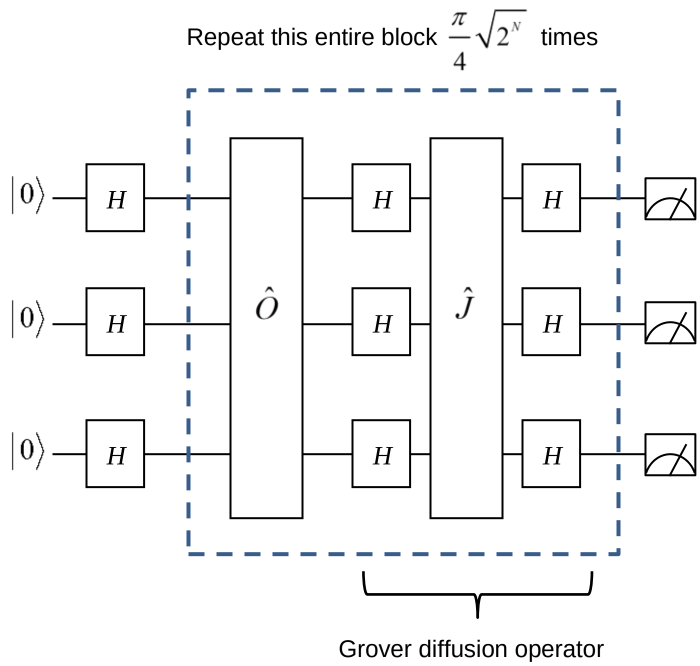


Fig. 4
Quantum circuit diagram for Grover's search algorithm.

Therefore the quantum oracle matrix is like the identity matrix, except that it has a –1 as the diagonal element for the correct question. When that question is 110, the matrix is

$$\hat{O} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

since $|110\rangle$ is the seventh basis vector. The oracle is unitary, when multiplied by its adjoint it gives the identity matrix. The operator $\hat{J}$ is like the oracle except the –1 element is always in the first position:

$$\hat{J} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The first set of Hadamard gates creates an equal superposition of all 2^N basis states. This superposition is passed to the oracle, essentially asking all 2^N questions at the same time. The oracle flips the sign of the amplitude for the correct question. The "Grover diffusion operator" is designed to convert this *phase* difference, which is unmeasurable, into a *magnitude* difference that will show up when the qubits are measured. It might seem better to use an oracle that encodes its output in a directly useable form such as,

$$\hat{O}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

If $\hat{O}'$ is applied to a superposition of all basis states, the result is immediately the basis state for the correct question i.e. $|110\rangle$. However the matrix $\hat{O}'$ is not unitary and therefore does not represent a possible operation in a quantum computer.

Implement Grover's Quantum Search Algorithm

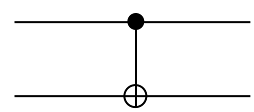
Implement the Grover search algorithm for $N = 3$ qubits as shown (in Fig. 4.). The necessary matrices are given above. The optimum number of repetitions should be close to $\frac{\pi}{4}\sqrt{2^N}$ but must be an integer.

- Set the oracle so the correct question is 110 as above, and with the optimum number of repetitions of the oracle plus Grover diffusion block. Run the computation many times. The measured result should be $|110\rangle$ more than 90% of the time.
- Change the oracle so a different question is correct.

- Change the number of repetitions. The percentage of time that the result is correct should decrease with either more or fewer repetitions.
- Generate a table for each question that shows the number of times the search was performed (loops through the search algorithm shown in the figure). The number of times $|\psi\rangle$ was measured and the results for each basis state.

Project 4: Gates operating on more than one qubit–Control gates

In this project we are going to use control gates. In particular a controlled NOT gate. The gate is represented by the circuit shown here. The solid circle shows the control qubit and the hollow circle represents the target qubit. This gate does the following: if the control qubit is in state $|0\rangle$ it does nothing to the target, if the control qubit is in state $|1\rangle$ it flips the target qubit–from $|0\rangle \rightarrow |1\rangle$ or $|1\rangle \rightarrow |0\rangle$.

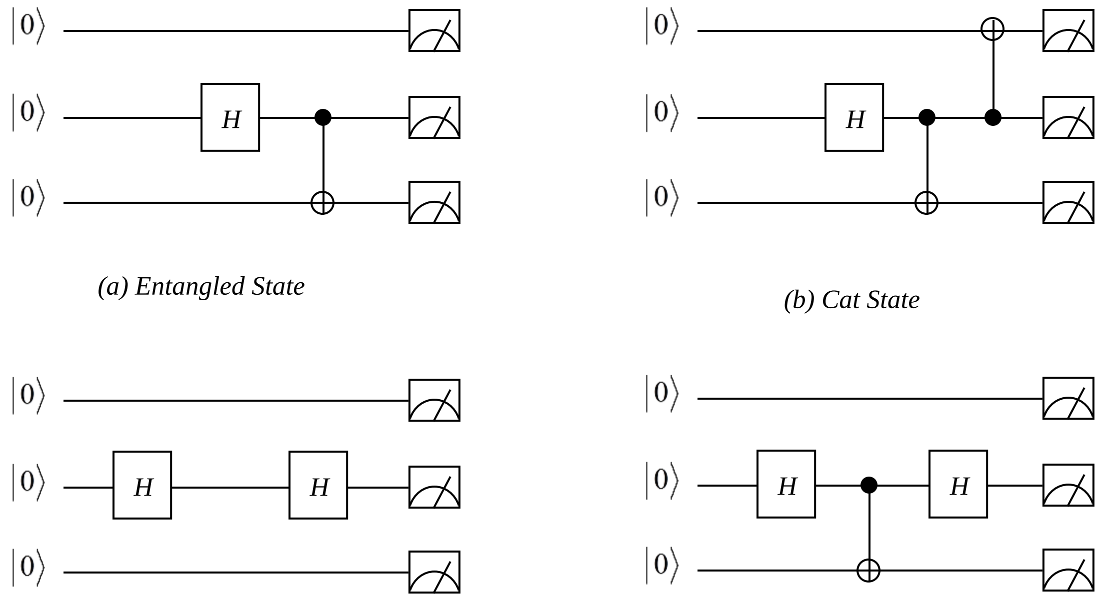


- Write code to run the four quantum computations shown below.
- Explain what you expect to measure.
- Perform the measurements and reconcile with your expectations. Provide the data in a neat tabular form.
- You should also include the *.m files to show that this works. Those files should be well documented so it is clear what they are supposed to do.

Try the computations shown in the Fig. 5. As before gates are applied to qubits by successively operating on $|\psi\rangle$ from the left. Initialize $|\psi\rangle$ to $|000\rangle$. The top line is for operations on qubit 1, second line for operations on qubit 2, etc. Also add one more gate to circuit (d) so the circuit behaves exactly like (c).

(a) The operation is $\hat{C}_{NOT}^{(2,3)}\hat{H}^{(2)}|\psi\rangle$, first the $\hat{H}^{(2)}$ operator operates on $|\psi\rangle$, followed by the $\hat{C}_{NOT}^{(2,3)}$ operator. There is a Hadamard gate on qubit 2, before qubit 2 is used to control qubit 3. The Hadamard gate puts qubit 2 in a superposition of $|0\rangle$ and $|1\rangle$. When this superposition is used to control qubit 3, it too is in a superposition. When the computation is carried out multiple times, the output should vary randomly between $|000\rangle$ and $|011\rangle$. The measured values for qubits 2 and 3 are both random, but they are correlated with each other. This is called an entangled state of qubits 2 and 3.

(b) The operation is $\hat{C}_{NOT}^{(2,1)}\hat{C}_{NOT}^{(2,3)}\hat{H}^{(2)}|\psi\rangle$. The Hadamard gate puts qubit 2 in a superposition of $|0\rangle$ and $|1\rangle$. The two CNOTs then correlate qubit 3 and 1 with qubit 2.



(a) Entangled State *(b) Cat State*

(c) Unobserved Superposition *(d) Observed Superposition*

Fig. 5
Schematic representation of four quantum computations using control gates.

(c) The operation is $\hat{H}^{(2)}\hat{H}^{(2)}|\psi\rangle$. The Hadamard gate puts qubit 2 in a superposition of $|0\rangle$ and $|1\rangle$ then the second Hadamard gate puts it back into the pure state $|0\rangle$.

(d) The operation is $\hat{H}^{(2)}\hat{C}_{NOT}^{(2,3)}\hat{H}^{(2)}|\psi\rangle$.This is like part (c) except a CNOT is used to observe the state of qubit 2 when it is in a superposition of $|0\rangle$ and $|1\rangle$. Qubit 3 is flipped by the CNOT to agree with the state of qubit 2, and when qubit 3 is measured it reveals what qubit 2 was between the two Hadamard gates. Observing qubit 2 destroys the coherence of the superposition, so the second Hadamard gate cannot put qubit 2 back into the pure state $|0\rangle$.

(e) In Project 4d the state of qubit 2 is not really observed until the qubits are measured. Figure out how to undo the observation of qubit 2 before the qubits are measured by adding one more gate, and verify that with this modification the results are the same as project 4c namely that the result is always $|000\rangle$. Draw the new quantum circuit below.

Project 4d also illustrates a peculiarity of quantum computation. Classically the only effect of the CNOT would be on qubit 3. In a quantum calculation, the CNOT might be used specifically for its effect on qubit 2.

Assessment

This material was presented in a course on modern physics over two semesters; it is being further developed during the current semester. The prerequisite for the class is General Physics II the second semester general physics course taken by engineers and chemistry students. This course is a requirement for electrical engineers and is a technical elective for other engineering majors. It is offered once a year during the winter semester. In the winter of 2016, thirteen students were enrolled and 12 took the Quantum Mechanics Conceptual Survey 2.0 (QMCS) the following year 7 students were enrolled and took the assessment.[13] The assessment is a 12 question multiple-choice test that is considered quite difficult. Student scores are reported by the study's authors to be in the range of 50%-70%. Based on the authors recommendations it was used only as a post-test. Student's prior knowledge of quantum mechanics is so lacking, that there is no baseline knowledge to assess.

During both terms Mermin's device was used to introduce quantum mechanics and provide a motivation for its study. During the winter of 2016 the MATLAB programing projects were optional and if students turned them in, their scores would replace any lower scores on other homework assignments. During the winter of 2017 all of the programming projects were assigned and the scores tabulated as part of the homework portion of the student's grade. During the winter of 2016 the average percentage of correct student responses was 46 while the average percentage of correct student responses was 47 for the winter 2017 cohort. The small sample size indicates that the difference is, in no way, statistically significant. We do note that the scores did not decrease dramatically. It is our intention to continue to develop the quantum measurement simulations and to analyze the assessment results on a question-by-question basis.

Appendix A– Sample student MATLAB code

Function to measure the quantum state:

```
function state = measure(psi)
R = rand(1);
q = abs(psi(1))^2;
if R < q
    state = 1;
else
    q = q + abs(psi(2))^2;
    if R < q
        state = 2;
    else
        q = q + abs(psi(3))^2;
        if R < q
            state = 3;
        else
            q = q + abs(psi(4))^2;
            if R < q
                state = 4;
            else
                q = q + abs(psi(5))^2;
                if R < q
```

```
                        state = 5;
                    else
                        q = q + abs(psi(6))^2;
                        if R < q
                            state = 6;
                        else
                            q = q + abs(psi(7))^2;
                            if R < q
                                state = 7;
                            else
                                state = 8;
                            end
                        end
                    end
                end
            end
        end
    end
```

Function to generate the appropriate Hadamard gate

```
function H_out = calc_H(n)
% This function calculates the Hadamard gate for each qubit.
% The input n is the qubit to act upon.
H = (1/sqrt(2))*[1 1;...
                 1 -1];
I = [1 0;...
     0 1];
switch n
    case 1  % First qubit: H x I x I
        H_out = kron(H,I);
        H_out = kron(H_out,I);
    case 2  % Second qubit: I x H x I
        H_out = kron(I,H);
        H_out = kron(H_out,I);
    case 3  % Third qubit: I x I x H
        H_out = kron(I,H);
        H_out = kron(I,H_out);
end
end
```

Function to generate the phase-shift gate operating on a specific qubit and rotating a given amount.

```
function R = calc_rotation(n,theta)
I = [1 0;
     0 1];
R = [cos(theta) sin(theta);
     -sin(theta) cos(theta)];
switch n
    case 1
        R = kron(kron(R,I),I);
    case 2
        R = kron(kron(I,R),I);
    case 3
```

```
        R = kron(kron(I,I),R);
end
end
```

Appendix B – Quantum Operators – Hadamard and phase-shift gates with 3 qubits

$$\hat{H}^{(1)} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$$\hat{H}^{(2)} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}$$

$$\hat{H}^{(3)} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

$$\hat{R}_{\theta}^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\theta} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} \end{pmatrix}$$

$$\hat{R}_{\theta}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} \end{pmatrix}$$

$$\hat{R}_{\theta}^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} \end{pmatrix}$$

References

[1] Riedinger, R., Wallucks, A., Marinković, I., Löschnauer, C., Aspelmeyer, M., Hong, S., and Gröblacher, S. "Remote quantum entanglement between two micromechanical oscillators." Nature, **556**, 473–477 (April 2018); https://doi.org/10.1038/s41586-018-0036-z.

[2] MATLAB, The MathWorks, Inc., Natick, Massachusetts, United States.

[3] Choi, H. S. (2017, June), *Incorporating Quantum Technologies into Engineering Curriculum* Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/28513.

[4] Mermin, N.D.,"Bringing home the atomic world; Quantum mysteries for anybody." Am. J. Phys. **49**, (10) 940–943 (October 1981); https://doi.org/10.1119/1.12594.

[5] Candela, D.,"Undergraduate computational physics projects on quantum computing." Am. J. Phys. **83**, (8) 688–702 (August 2015); https://doi.org/10.1119/1.4922296.

[6] Einstein, A., Podolsky, B., and Rosen, N., "Can quantum mechanical description of physical reality be considered complete?" Phys. Rev. **47**, 770-780 (1935).

[7] Mermin, N.D.,"Quantum mysteries revisited." Am. J. Phys. **58**, (8) 731–734 (August 1990); https://doi.org/10.1119/1.16503.

[8] Mermin, N.D.,"Quantum mysteries refined." Am. J. Phys. **60**, (10) 880–887 (October 1994); https://doi.org/10.1119/1.17733.

[9] Schroeder, D.V., "Entanglement isn't just for spin." Am. J. Phys. **85**, (11) 812–820 (November 2017); https://doi.org/10.1119/1.5003808.

[10] Dirac, P. A. M., "A new notation for quantum mechanics". Mathematical Proceedings of the Cambridge Philosophical Society. **35** (3), 416–418, (1939).

[11] Dirac, P. A. M., The Principles of Quantum Mechanics, 4th ed., Oxford University Press (1958), pg. 16-22.

[12] Mermin, N.D.,"From Cbits to Qbits: Teaching computer scientists quantum mechanics." Am. J. Phys. **71**, (1) 23–30 (January 2003); https://doi.org/10.1119/1.1522741.

[13] McKagan, S.B.,Perkins, K.K., and Wieman, C.E., "Design and validation of the Quantum Mechanics Conceptual Survey." Phys. Rev. Special Topics – Physics education Research, **6**, (2010); https://doi.org/10.1103/PhysRevSTPER.6.020121.