QDENSITY - A MATHEMATICA QUANTUM COMPUTER SIMULATION

Bruno Juliá-Díaz [a,b], Joseph M. Burdis [a,c] and Frank Tabakin [a]

[a]*Department of Physics and Astronomy*
University of Pittsburgh
Pittsburgh, PA, 15260

[b]*DAPNIA, DSM, CEA/Saclay*
91191 Gif-sur-Yvette, France

[c]*Department of Mathematics*
North Carolina State University
Raleigh, NC 27695

Abstract

This Mathematica 5.2 package [1] is a simulation of a Quantum Computer. The program provides a modular, instructive approach for generating the basic elements that make up a quantum circuit. The main emphasis is on using the density matrix, although an approach using state vectors is also implemented in the package. The package commands are defined in *Qdensity.m* which contains the tools needed in quantum circuits, e.g. multiqubit kets, projectors, gates, etc. Selected examples of the basic commands are presented here and a tutorial notebook, *Tutorial.nb* is provided with the package (available on our website) that serves as a full guide to the package. Finally, application is made to a variety of relevant cases, including Teleportation, Quantum Fourier transform, Grover's search and Shor's algorithm, in separate notebooks: *QFT.nb*, *Teleportation.nb*, *Grover.nb* and *Shor.nb* where each algorithm is explained in detail. Finally, two examples of the construction and manipulation of cluster states, which are part of "one way computing" ideas, are included as an additional tool in the notebook *Cluster.nb*. A Mathematica palette containing most commands in QDENSITY is also included: *QDENSpalette.nb* .

---

[1] QDENSITY is available at http://www.pitt.edu/˜tabakin/QDENSITY

Program Summary

Title of program: QDENSITY
Catalogue identifier:
Program summary URL: http://cpc.cs.qub.ac.uk/summaries
Program available from: CPC Program Library, Queen's University of Belfast, N. Ireland
Operating systems: Any which supports Mathematica; tested under Microsoft Windows XP, Macintosh OS X, and Linux FC4.
Programming language used: Mathematica 5.2
Number of bytes in distributed program, including test code and documentation: 1185886
Distribution format: tar.gz
Nature of Problem: Analysis and design of quantum circuits, quantum algorithms and quantum clusters.
Method of Solution: A Mathematica package is provided which contains commands to create and analyze quantum circuits. Several Mathematica notebooks containing relevant examples: Teleportation, Shor's Algorithm and Grover's search are explained in detail. A tutorial, Tutorial.nb is also enclosed.

Contents

1   INTRODUCTION

There is already a rich Quantum Computing (QC) literature [1] which holds forth the promise of using quantum interference and superposition to solve otherwise intractable problems. The field has reached the point that experimental realizations are of paramount importance and theoretical tools towards that goal are needed: to gauge the efficacy of various approaches, to understand the construction and efficiency of the basic logical gates, and to delineate and control environmental decoherence effects.

In this paper, a Mathematica [2] package provides a simulation of a Quantum Computer that is both flexible and an improvement over earlier such works [3]. It is a bona fide simulation in that its success depends on quantum interference and superposition and is not just a simulation of the QC experience. The flexibility is generated by a modular approach to all of the initializations, operators, gates, and measurements, which then can be readily used to describe the basic QC Teleportation [4], Grover's search [5,6] and Shor's factoring [7] algorithms. We also adopt a density matrix approach as an organizational framework for introducing fundamental Quantum Computing concepts in a manner that allows for more general treatments, such as handling the dynamics stipulated by realistic Hamiltonians and including environmental effects. That approach allows us to invoke many of the dynamical theories based on the time evolution of the density matrix. Since much of the code uses the

density matrix, we call it "QDENSITY," which stands for Quantum computing with a density matrix framework. However, the code also provides the tools to work directly with multi-qubit states as an alternative to the density matrix description.

In section 2, we introduce one qubit state vectors and associated spin operators, including rotations, and introduce commands from QDENSITY. The basic characteristics of the density matrix are then discussed in a pedagogic manner in section 3. Then in section 4, methods for handling multi-qubit operators, density matrices, and state vectors with commands from QDENSITY are presented. Also in that section, we show how to take traces, subtraces and how to combine subtraces with general projection operators to simulate projective measurements in a multi-qubit context. The basic one, two and three qubit gates (Hadamard, CNOT, CPHASE, Toffoli, etc.) needed for the QC circuits are shown in section 5. The production of entangled states, such as the two-qubit Bell [8] states, the three-qubit GHZ [9] states, among others, [10] are illustrated in both density matrix and state vector renditions in section 6. In sections 7-9, Teleportation, Grover's search, and Shor's factoring algorithms are outlined, with the detailed instructions relegated to associated notebooks. Sample application to the cluster or "one-way computing" model of QC is presented in section 10. Possible future applications of QDENSITY are given in the conclusion section 11.

The basic logical gates used in the circuit model of Quantum Computing are presented in a way that allows ease of use and hence permits one to construct the unitary operators corresponding to well-know quantum algorithms. These algorithms are developed explicitly in the Mathematica notebooks as a demonstration of the application of QDENSITY. A tutorial notebook (Tutorial.nb) available on our web site guides the user through the requisite manipulations. Many examples from QDENSITY, as defined in the package file Qdensity.m, are discussed throughout the text, which hopefully, with the tutorial notebook, will help the user to employ this tool.

All these examples are instructive in two ways. One way is to learn how to handle QDENSITY for other future applications and generalizations, such as studying entanglement measures, examining the time evolution generated by experiment-based realistic Hamiltonians, error correction methods, and the role of the environment and its affect on coherence. Thus the main motivation for emphasizing a density matrix formulation is that the time evolution can be described, including effects of an environment, starting from realistic Hamiltonians [11,12]. Therefore, QDENSITY provides an introduction to methods that can be generalized to an increasingly realistic description of a real quantum computer.

Another instructive feature is to gain insight into how quantum superposition

5

and interference are used in QC to pose and to answer questions that would be inaccessible using a classical computer. Thus we can form an initial description of a quantum multi-qubit state and have it evolve by the action of carefully designed unitary operators. In that development, the prime characteristics of superposition and interference of probability amplitudes is cleverly applied in Quantum Computing to enhance the probability of getting the right answer to problems that would otherwise take an immense time to solve.

In addition to applying QDENSITY to the usual quantum circuit model for QC, we have adapted it to examine the construction of cluster states and the steps needed to reproduce general one qubit and two qubit operations. These cluster model examples are included to open the door for future studies of the quite promising cluster model [13] or "one-way computing" approach for QC.

We sought to simulate as large a system of qubits as possible, using new features of Mathematica. Of course, this code is a simulation of a quantum computer based on Mathematica code run on a classical computer. So it is natural that the simulation saturates memory for large qubit spaces; after all, if the QC algorithms always worked efficiently on a classical computer there would be no need for a quantum computer.

Throughout the text, sample QDENSITY commands are presented in sections called "Usage." The reader should consult Tutorial.nb for more detailed guidance.

2   ONE QUBIT SYSTEMS

The state of a quantum system is described by a wave function which in general depends on the space or momentum coordinates of the particles and on time. In Dirac's representation independent notation, the state of a system is a vector in an abstract Hilbert space $\mid \Psi(t) >$, which depends on time, but in that form one makes no choice between the coordinate or momentum space representation. The transformation between the space and momentum representation is contained in a transformation bracket. The two representations are related by Fourier transformation, which is the way Quantum Mechanics builds localized wave packets. In this way, uncertainty principle limitations on our ability to measure coordinates and momenta simultaneously with arbitrary precision are embedded into Quantum Mechanics (QM). This fact leads to operators, commutators, expectation values and, in the special cases when a physical attribute can be precisely determined, eigenvalue equations with Hermitian operators. That is the content of many quantum texts. Our purpose is now to see how to define a density matrix, to describe systems with two degrees of freedom as needed for quantum computing.

Spin, which is the most basic two-valued quantum attribute, is missing from a spatial description. This subtle degree of freedom, whose existence is deduced by analysis of the Stern-Gerlach experiment, is an additional Hilbert space vector feature. For example, for a single spin $1/2$ system the wave function including both space and spin aspects is:

$$\Psi(\vec{r}_1, t) \mid s \; m_s >, \tag{1}$$

where $\mid s \; m_s >$ denotes a state that is simultaneously an eigenstate of the particle's total spin operator $s^2 = s_x^2 + s_y^2 + s_z^2$, and of its spin component operator s_z. That is

$$s^2 \mid sm_s >= \hbar^2 s(s+1) \mid sm_s > \qquad s_z \mid sm_s >= \hbar m_s \mid sm_s > . \tag{2}$$

For a spin $1/2$ system, we denote the spin up state as $\mid sm_s >\rightarrow \mid \frac{1}{2}, \frac{1}{2} >\equiv\mid 0 >$, and the spin down state as $\mid sm_s >\rightarrow \mid \frac{1}{2}, -\frac{1}{2} >\equiv\mid 1 >$.

We now arrive at the definition of a one qubit state as a superposition of the two states associated with the above 0 and 1 bits:

$$\mid \Psi >= a \mid 0 > +b \mid 1 >, \tag{3}$$

where $a \equiv< 0 \mid \Psi >$ and $b \equiv< 1 \mid \Psi >$ are complex probability amplitudes for finding the particle with spin up or spin down, respectively. The normalization of the state $< \Psi \mid \Psi >= 1$, yields $\mid a \mid^2 + \mid b \mid^2 = 1$. Note that the spatial aspects of the wave function are being suppressed; which corresponds to the particles being in a fixed location, such as at quantum dots. [2]

An essential point is that a QM system can exist in a superposition of these two bits; hence, the state is called a quantum-bit or "qubit." Although our discussion uses the notation of a system with spin, it should be noted that the same discussion applies to any two distinct states that can be associated with $\mid 0 >$ and $\mid 1 >$. Indeed, the following section on the Pauli spin operators is really a description of any system that has two recognizable states.

2.0.1 Usage

QDENSITY includes commands for qubit states as ket and bra vectors. For example, commands Ket$[0]$, Bra$[0]$,Ket$[1]$, Bra$[1]$, yield

---

[2] When these separated systems interact, one might need to restore the spatial aspects of the full wave function.

$$\text{In}[1] := \mathbf{Ket}[\mathbf{0}]$$

$$\text{Out}[1] := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\text{In}[2] := \mathbf{Ket}[\mathbf{1}]$$

$$\text{Out}[2] := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\text{In}[3] := \mathbf{Bra}[\mathbf{0}]$$
$$\text{Out}[3] := (\mathbf{1} \quad \mathbf{0})$$

$$\text{In}[4] := \mathbf{Bra}[\mathbf{1}]$$
$$\text{Out}[4] := (\mathbf{0} \quad \mathbf{1})$$

These are the computational basis states, i.e. eigenstates of the spin operator in the z-direction.

States that are eigenstates of the spin operator in the x-direction are invoked by the commands

$$\text{In}[5] := \mathbf{BraX}[\mathbf{0}]$$

$$\text{Out}[5] := \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \quad \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \right)$$

which is equivalent to:

$$\text{In}[6] := (\mathbf{Bra}[\mathbf{0}] + \mathbf{Bra}[\mathbf{1}])/\sqrt{2}$$

$$\text{Out}[6] := \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \quad \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \right)$$

Eigenstates of the spin operator in the y-direction are invoked similarly the commands $\mathbf{BraY}[\mathbf{0}]$, $\mathbf{BraY}[\mathbf{1}]$, etc.

We use the case of a spin $1/2$ particle to describe a quantum system with two discrete levels; the same description can be applied to any QM system with two distinct levels. The spin $\vec{s}$ operator is related to the three Pauli spin operators $\sigma_x, \sigma_y, \sigma_z$ by

$$\vec{s} \equiv (\frac{\hbar}{2})\vec{\sigma}, \tag{4}$$

from which we see that $\vec{\sigma}$ is an operator that describes the spin $1/2$ system in units of $\frac{\hbar}{2}$. Since spin is an observable, it is represented by a Hermitian operator, $\vec{\sigma}^{\dagger} = \vec{\sigma}$. We also know that measurement of spin is subject to the uncertainty principle, which is reflected in the non-commuting operator properties of spin and hence of the Pauli operators. For example, from the standard commutator property for any spin $[s_x, s_y] = i\hbar s_z$, one deduces that the Pauli operators do not commute

$$[\sigma_x, \sigma_y] = 2i\sigma_z \ . \tag{5}$$

This holds for all cyclic components so we have a general form [3]

$$[\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k \ . \tag{6}$$

An important property of the spin algebra is that the total spin commutes with any component of the spin operator $[s^2, s_i] = 0$ for all i. The physical consequence is that one can simultaneously measure the total spin of the system and one component (usually s_z) of the spin. Only one component is a candidate for simultaneous measurement because the commutator $[s_x, s_y] = i\hbar s_z$ is already an uncertainty principle constraint on the other components. As a result of the ability to measure s^2 and s_z simultaneously, the allowed states of the spin $1/2$ system are restricted to being spin-up and spin-down with respect to a specified fixed direction $\hat{z}$, called the axis of quantization. States defined relative to that fixed axis are called "computational basis" states, in the QC jargon. The designation arises because as noted already one can identify spin-up with a state $\mid 0 >$, which designates the digit or bit 0, and a spin-down state as $\mid 1 >$, which designates the digit or bit 1.

The fact that there are just two states (up and down) also implies properties of the Pauli operators. We construct [4] the raising and lowering operators

---

[3] Here the Levi-Civita symbol is nonzero only for cyclic order of components $ijk = xyz, yzx, zxy$, for which $\epsilon_{ijk} = 1$. For anti-cyclic order of components $ijk = xzy, zyx, yxz$ $\epsilon_{ijk} = -1$. It is otherwise zero.

[4] With the definition $s_{\pm} \equiv s_x \pm i s_y$, and using the original spin commutation rules, it follows that $[s_{\pm}, s_z] = \mp s_{\pm}$, which reveals that $s_{\pm}$ and hence also $\sigma_{\pm}$ are raising

$\sigma_\pm = \sigma_x \pm i\sigma_y$, and note that the raising and lower process is bounded

$$\sigma_+ \mid 0 >= 0 \qquad \sigma_- \mid 1 >= 0. \tag{7}$$

Hence, raising a two-valued state up twice or lowering it twice yields a null (unphysical) Hilbert space; this property tells us additional aspects of the Pauli operator. Since

$$\sigma_\pm \sigma_\pm = (\sigma_x \pm i\sigma_y)^2 = \sigma_x^2 - \sigma_y^2 \pm (\sigma_x\sigma_y + \sigma_y\sigma_x) = 0, \tag{8}$$

we deduce that $\sigma_x^2 = \sigma_y^2$, and that the anti-commutator

$$\{\sigma_x, \sigma_y\} \equiv \sigma_x\sigma_y + \sigma_y\sigma_x = 0. \tag{9}$$

The anti-commutation property is thus a direct consequence of the restriction to two levels.

The spin 1/2 property is often expressed as: $s^2 \mid sm_s >= \hbar^2 s(s+1) \mid sm_s >= \frac{3}{4}\hbar^2 \mid sm_s >= \frac{\hbar^2}{4}\sigma^2 \mid sm_s >$. We have $\sigma^2 = 3 = \sigma_x^2 + \sigma_y^2 + \sigma_z^2 = 2\sigma_x^2 + 1$, where we use the above equality $\sigma_x^2 = \sigma_y^2$, and from the $\hat{z}$ eigenvalue equation the property $\sigma_z^2 = 1$, to deduce that

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 1. \tag{10}$$

Another useful property of the Pauli matrices is obtained by combining the above properties and commutator and anti-commutator into one expression for a given spin 1/2 system

$$\sigma_i\sigma_j = \delta_{ij} + i\epsilon_{ijk}\sigma_k, \tag{11}$$

where indices i, j, k take on the values x, y, z, and repeated indices are assumed to be summed over. For two general vectors, this becomes

$$(\vec{\sigma} \cdot \vec{A})(\vec{\sigma} \cdot \vec{B}) = \vec{A} \cdot \vec{B} + i(\vec{A} \times \vec{B}) \cdot \vec{\sigma}. \tag{12}$$

For $\vec{A} = \vec{B} = \vec{\eta}$, a unit vector $(\vec{\sigma} \cdot \vec{\eta})^2 = 1$, which will be useful later.

These operator properties can also be represented by the Pauli-spin matrices, where we identify the matrix elements by

$$< s\, m'_s \mid \sigma_z \mid s\, m_s > \longrightarrow \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{13}$$

and lowering operators. The general result, including the limit on the total spin is
$s_\pm \mid s\, m_s >= \sqrt{s(s+1) - m_s(m_s \pm 1)} \mid s\, m_s \pm 1 >.$

Similarly for the $x-$ and $y-$component spin operators

$$< sm'_s \mid \sigma_x \mid s\, m_s > \longrightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad < sm'_s \mid \sigma_y \mid sm_s > \longrightarrow \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$
(14)

These are all Hermitian matrices $\sigma_i = \sigma_i^\dagger$.

Also, the matrix realization for the raising and lowering operators are:

$$\sigma_+ = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} \qquad \sigma_- = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}.$$
(15)

Here $\sigma_+^\dagger = \sigma_-$.

Note that these 2×2 Pauli matrices are traceless $\mathrm{Tr}[\vec{\sigma}] = 0$, unimodular $\sigma_i^2 = 1$ and have unit determinant $\mid \det \sigma_i \mid = 1$. Along with the unit operator

$$\sigma_0 = \mathbf{1} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$
(16)

the four Pauli operators form a basis for the expansion of any spin operator in the single qubit space. For example, we can express the rotation of a spin as such an expansion and later we shall introduce a density matrix for a single qubit in the form $\rho = a + \vec{b} \cdot \vec{\sigma} = a + b\, \vec{n} \cdot \vec{\sigma}$ to describe an ensemble of particle spin directions as occurs in a beam of spin-1/2 particles.

In QDENSITY, we denote the four matrices by σ_i where $i = 0$ is the unit matrix and $i = 1, 2, 3$ corresponds to the components $x, y,$ and z. To produce the Pauli spin operators in QDENSITY, one can use either the Greek form or the expression $s[i]$.

2.1.1 Usage

QDENSITY includes commands for the Pauli operators. For example, there are three equivalent ways to invoke the Pauli σ_y matrix in QDENSITY:

$$\mathrm{In}[7] := \sigma_\mathbf{y}$$

$$\mathrm{Out}[7] := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\text{In}[8] := \mathbf{s[2]}$$

$$\text{Out}[8] := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

The third way is to use the commands **Sigma0**, **Sigma1**, **Sigma2**, or **Sigma3**.

Note that

$$\text{In}[9] := \sigma_{\mathbf{2}} \, . \, \mathbf{KetY[0]} - \mathbf{KetY[0]}$$

$$\text{Out}[9] := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and

$$\text{In}[10] := \sigma_{\mathbf{2}} \, . \, \mathbf{KetY[1]} + \mathbf{KetY[1]}$$

$$\text{Out}[10] := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

confirm that **KetY[0]** and **KetY[1]** are indeed eigenstates of σ_y. Note that the $\cdot$ is used to take the dot product.

2.2   Pauli Operators in Hilbert Space

It is often convenient to express the above matrix properties in the form of operators in Hilbert space. For a general operator Ω, using closure, we have

$$\Omega = \sum_n \sum_{n'} | \, n > < n \, | \, \Omega \, | \, n' > < n' \, | \; . \tag{17}$$

For the four Pauli operators this yields:

$$
\begin{aligned}
\sigma_0 &= | \, 0 > < 0 \, | + | \, 1 > < 1 \, | \\
\sigma_1 &= | \, 0 > < 1 \, | + | \, 1 > < 0 \, | \\
\sigma_2 &= -i \, | \, 0 > < 1 \, | + i \, | \, 1 > < 0 \, | \\
\sigma_z &= | \, 0 > < 0 \, | - | \, 1 > < 1 \, | \; .
\end{aligned}
\tag{18}
$$

Taking matrix elements of these operators reproduces the above Pauli matrices. Also note we have the general trace property

$$\text{Tr} \mid a >< b \mid = \sum_n < n \mid a >< b \mid n > = \sum_n < b \mid n >< n \mid a > = < b \mid a >,$$

(19)

where $\mid n >$ is a complete orthonormal (CON) basis [5]. Applying this trace rule to the above operator expressions confirms that $\text{Tr}[\sigma_x] = \text{Tr}[\sigma_y] = \text{Tr}[\sigma_z] = 0$, and $\text{Tr}[\sigma_0] = 2$.

Another useful trace rule is $\text{Tr}[\, \Omega \mid a >< b \mid] = < b \mid \Omega \mid a >$.

2.3   Rotation of Spin

Another way to view the above superposition, or qubit, state is that a state originally in the $\hat{z}$ direction $\mid 0 >$, has been rotated *to a new direction* specified by a unit vector $\hat{n} = (n_x, n_y, n_z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$, as shown in Fig. 1.
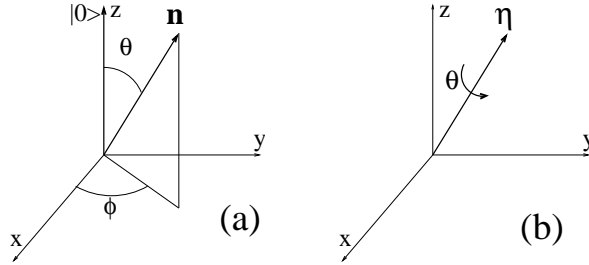
Fig. 1. Active rotation *to* a direction $\hat{n}$ (a); and active rotation *around* a vector $\hat{\eta}$ (b).

The rotated spin state

$$\mid \hat{n} > = \cos(\theta/2)e^{-i\phi/2} \mid 0 > + \sin(\theta/2)e^{+i\phi/2} \mid 1 > = \begin{pmatrix} \cos(\theta/2)e^{-i\phi/2} \\ \sin(\theta/2)e^{+i\phi/2} \end{pmatrix},$$

(20)

is a normalized eigenstate of the operator

$$\vec{\sigma} \cdot \hat{n} = \begin{pmatrix} n_z & n_x - in_y \\ n_x + in_y & -n_z \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta e^{-i\phi} \\ \sin\theta e^{i\phi} & -\cos\theta \end{pmatrix}.$$

(21)

We see that

$$\vec{\sigma} \cdot \hat{n} \mid \hat{n} > = \mid \hat{n} > .$$

(22)

---

[5] For a CON basis, we have closure $\sum_n \mid n >< n \mid = \mathbf{1}$.

13

The half angles above reflect the so-called spinor nature of the QM state of a spin 1/2 particle in that a double full rotation is needed to bring the wave function back to its original form.

These rotated spin states allow us to pick a different axis of quantization $\hat{n}$ for each particle in an ensemble of spin 1/2 systems. These states are normalized $< \hat{n} \mid \hat{n} > = 1$, but are not orthogonal $< \hat{n}' \mid \hat{n} > \neq 1$, when the $\hat{n}$ angles θ, ϕ do not equal the $\hat{n}'$ angles θ', ϕ'.

Special cases of the above states with spin pointing in the directions $\pm\hat{x}$ and $\pm\hat{y}$ are within phases:

$$\mid \pm x > = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix} \rightarrow \frac{\mid 0 > \pm \mid 1 >}{\sqrt{2}},$$

$$\mid \pm y > = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i \end{pmatrix} \rightarrow \frac{\mid 0 > \pm i \mid 1 >}{\sqrt{2}}. \tag{23}$$

Hilbert space versions are also shown above.

Rotation can also be expressed as a rotation of an initial spin-up system *about a rotation axis* $\hat{\eta}$ by an angle γ. Thus an operator $R_\gamma \equiv e^{-i\frac{\gamma}{2}\vec{\sigma}\cdot\hat{\eta}}$, acting as

$$\mid \Psi > = R_\gamma \mid 0 > \tag{24}$$

can also rotate the spin system state to new direction. This rotation operator can be expanded in the form

$$R_\gamma = e^{-i\frac{\gamma}{2}\vec{\sigma}\cdot\hat{\eta}} = \cos\frac{\gamma}{2}\,\sigma_0 - i\sin\frac{\gamma}{2}\,\vec{\sigma}\cdot\hat{\eta}, \tag{25}$$

which follows from the property that $(\vec{\sigma}\cdot\hat{\eta})^2 = \hat{\eta}\cdot\hat{\eta} + i(\hat{\eta}\times\hat{\eta})\cdot\vec{\sigma} = 1$. A special case of the state generated by this rotation $R_\gamma \mid 0 >$ is a $\gamma = \pi/2$ rotation about the $\hat{\eta} \rightarrow \hat{y}$ axis. Then the rotation operator is

$$R_{\pi/2} = e^{-i\frac{\pi}{4}\sigma_y} = \cos\frac{\pi}{4}\sigma_0 - i\sin\frac{\pi}{4}\,\sigma_y\ . \tag{26}$$

Introducing the Pauli matrices, this becomes

$$R_{\pi/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}. \tag{27}$$

This rotation about an axis again yields the same result; namely,

$$R_{\pi/2} \mid 0 > = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \qquad (28)$$

Similar steps apply for a rotation about the $\hat{x}$ axis by $\gamma = \pi/2$, which yields the earlier $\mid \pm y >$ states.

From normalization of the rotated state $< \Psi \mid \Psi > = < 0 \mid R_\gamma^\dagger R_\gamma \mid 0 > = < 0 \mid 0 > = 1$, we see that the rotation is a unitary $R_\gamma^\dagger R_\gamma = \mathbf{1}$ operator.

2.3.1 Usage

The trace of the Pauli operators is invoked in QDENSITY by:

$$\mathrm{In}[1] := \mathbf{Tr}[\sigma_\mathbf{2}]$$
$$\mathrm{Out}[1] := \mathbf{0}$$

Rotation about the X axis is represented by

$$\mathrm{In}[2] := \mathbf{RotX}[\theta]$$
$$\mathrm{Out}[2] := \begin{pmatrix} \mathrm{Cos}\left[\frac{\theta}{2}\right] & -i\mathrm{Sin}\left[\frac{\theta}{2}\right] \\ -i\mathrm{Sin}\left[\frac{\theta}{2}\right] & \mathrm{Cos}\left[\frac{\theta}{2}\right] \end{pmatrix}$$

Commands for other directions are described in *Tutorial.nb*; see RotX[θ], RotY[θ], Rotqbit[vec,θ].

2.4 One Qubit Projection

For a one qubit system, it is simple to define operators that project on to the spin-up or spin-down states. These projection operators are:

$$\mathcal{P}_0 \equiv \mid 0 > < 0 \mid \qquad \mathcal{P}_1 \equiv \mid 1 > < 1 \mid. \qquad (29)$$

These are Hermitian operators, and by virtue of closure, sum to one $\sum_{a=0,1} \mathcal{P}_a = 1$. They can also be expressed in terms of the σ_z operator as

$$\mathcal{P}_0 = \frac{1 + \sigma_z}{2} \qquad \mathcal{P}_1 = \frac{1 - \sigma_z}{2}, \tag{30}$$

or in matrix form

$$\mathcal{P}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad \mathcal{P}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{31}$$

One can also project to other directions. For example, projection of a qubit on to the $\pm\hat{x}$ or $\pm\hat{y}$ directions involves the projection operators

$$\mathcal{P}_{\pm x} = \mid \pm\hat{x} >< \pm\hat{x} \mid = \frac{1 \pm \sigma_x}{2},$$
$$\mathcal{P}_{\pm y} = \mid \pm\hat{y} >< \pm\hat{y} \mid = \frac{1 \pm \sigma_y}{2}. \tag{32}$$

2.4.1 Usage

The above projections operators are invoked in QDENSITY by:

$$\mathrm{In}[1] := \mathcal{P_0}$$

$$\mathrm{Out}[1] := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathrm{In}[2] := \mathcal{P_1}$$

$$\mathrm{Out}[2] := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Projection operators using the x-basis are invoked by:

$$\mathrm{In}[3] := \mathbf{PX[0]}$$

$$\mathrm{Out}[3] := \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$\text{In}[4] := \mathbf{PX[1]}$$

$$\text{Out}[4] := \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

A general operator **ProjG[a,vec]** to project into a direction stipulated by a unit three vector **vec**, with a=0 or 1, is also provided. See *Tutorial.nb* for more examples. These operators are useful for projective measurements.

3 THE (SPIN) DENSITY MATRIX

The above spin rotated wave functions can be used to obtain the expectation value of some relevant Hermitian operator $\Omega = \Omega^\dagger$, which represents a physical observable. Let us assume that a system is in a state labelled by α with a state vector $\mid \alpha >$. In general the role of the label α could be to denote a spatial coordinate (or a momentum), if we were considering an ensemble of localized particles. For the spin density matrix, we use α to label the various spin directions $\hat{n}$.

The average or expectation value of a general observable Ω is then $< \alpha \mid \Omega \mid \alpha >$. This expectation value can be interpreted simply by invoking eigenstates of the operator Ω

$$\Omega \mid \nu >= \omega_\nu \mid \nu >, \tag{33}$$

where ω_ν are real eigenvalues and $\mid \nu >$ are the eigenstates, which are usually a complete orthonormal basis (CON). The physical meaning of the eigenvalue equation is that if the system is in the eigenstate $\mid \nu >$, there is no uncertainty $\Delta\Omega$ in determining the eigenvalue, e.g.

$$(\Delta\Omega)^2 \equiv < \nu \mid \Omega^2 \mid \nu > - < \nu \mid \Omega \mid \nu >^2 = \omega_\nu^2 - \omega_\nu^2 \equiv 0. \tag{34}$$

Using the eigenstates $\mid \nu >$, we can now see the basic meaning of the expectation value, which is a fundamental part of QM. The eigenstates form a CON basis. That means any function can be expanded in this basis and that the coefficients can be obtained by an overlap integral. For example, in general terms the completeness (C) allows the expansion

$$\mid \Psi >= \sum_\nu c_\nu \mid \nu > . \tag{35}$$

The OrthoNormal (ON) aspect is $< \nu \mid \nu' >= \delta_{\nu\nu'}$. Thus

$$< \nu' \mid \Psi >= \sum_\nu c_\nu \ < \nu' \mid \nu > \ = c_{\nu'}, \tag{36}$$

reinserting this yields

$$| \Psi > = \sum_\nu < \nu | \Psi > | \nu > = \sum_\nu | \nu > < \nu | \ \Psi > = \mathbf{I} | \Psi > .$$

Thus we see completeness with orthonormality of the basis can be expressed in the closure form

$$\sum_\nu | \nu > < \nu | = \mathbf{I}, \tag{37}$$

with $\mathbf{I}$ the unit operator in the Hilbert space.

With closure (e.g. a CON basis), we can now see that the expectation value breaks in to a sum of the form

$$
\begin{aligned}
< \alpha | \Omega | \alpha > &= \sum_\nu \sum_{\nu'} < \alpha | \nu > < \nu | \Omega | \nu' > < \nu' | \alpha > \\
&= \sum_\nu \omega_\nu < \nu | \alpha > < \alpha | \nu > = \sum_\nu \omega_\nu P_\nu^\alpha.
\end{aligned}
$$

Here $P_\nu^\alpha = < \nu | \alpha > < \alpha | \nu > = |< \nu | \alpha >|^2$ is the positive real probability of the state $| \alpha >$ being in the eigenstate $| \nu >$. Hence we see that the quantum average or expectation value is a sum of that probability times the associated eigenvalue ω_ν over all possible values ν. That is the characteristic of a quantum average.

As the next step towards the spin density matrix, consider the case that we have an ensemble of such quantum systems. Each system is considered not to have quantum interference with the other members of the ensemble. That situation can be realized by the ensemble being located at separate sites with non-overlapping localized wave packets and also in the case of a low density beam, i.e. separated particles in the beam. This allows us to take a classical average over the ensemble.

Suppose that the first member of the ensemble is produced in the state $| \alpha >$, the next in $| \alpha' >$, etc. The ensemble average is then a simple classical average

$$< \Omega > = \frac{\sum_\alpha < \alpha | \Omega | \alpha > \mathbf{P}_\alpha}{\sum_\alpha \mathbf{P}_\alpha}, \tag{38}$$

where $\mathbf{P}_\alpha$ is the probability that a particular state α appears in the ensemble. Summing over all possible states of course yields $\sum_\alpha \mathbf{P}_\alpha = 1$. The above expression is a combination of a classical ensemble average with the quantum mechanical expectation value. It contains the idea that each member of the ensemble interferes only with itself quantum mechanically and that the ensemble involves a simple classical average over the probability distribution of the ensemble.

18

We are close to introducing the density matrix. This is implemented by using closure and rearranging. Consider

$$\sum_\alpha <\alpha\mid\Omega\mid\alpha> \mathbf{P}_\alpha = \sum_\alpha \sum_{mm'} <\alpha\mid m><m\mid\Omega\mid m'><m'\mid\alpha>\ \mathbf{P}_\alpha,$$

(39)

where $\mid m>$ denotes any CON basis. Now rearrange the above to

$$\sum_\alpha <\alpha\mid\Omega\mid\alpha> \mathbf{P}_\alpha = \sum_{mm'} \sum_\alpha <m'\mid\alpha><\alpha\mid m>\ \mathbf{P}_\alpha <m\mid\Omega\mid m'>,$$

(40)

and then define the **density operator** by

$$\rho \equiv \sum_\alpha \mid\alpha><\alpha\mid \mathbf{P}_\alpha$$

(41)

and the associated density matrix in the CON basis $\mid m>$ as $<m\mid\rho\mid m'>=$ $\sum_\alpha <m\mid\alpha><\alpha\mid m'> \mathbf{P}_\alpha$. We often refer to either the density operator or the density matrix simply as the "density matrix," albeit one acts in the Hilbert space and the other is an explicit matrix. The ensemble average can now be expressed as a ratio of traces [6]

$$<\Omega> = \frac{\mathrm{Tr}[\rho\Omega]}{\mathrm{Tr}[\rho]},$$

(42)

which entails the properties that

$$\mathrm{Tr}[\rho] = \sum_m <m\mid\rho\mid m> = \sum_\alpha \mathbf{P}_\alpha \sum_m <\alpha\mid m><m\mid\alpha>$$
$$= \sum_\alpha \mathbf{P}_\alpha <\alpha\mid\alpha> = \sum_\alpha \mathbf{P}_\alpha = 1,$$

(43)

and

$$\mathrm{Tr}[\rho\Omega] = \sum_{mm'} <m\mid\rho\mid m'><m'\mid\Omega\mid m>$$
$$= \sum_\alpha \sum_{mm'} \mathbf{P}_\alpha <\alpha\mid m'><m'\mid\Omega\mid m><m\mid\alpha>$$
$$= \sum_\alpha \mathbf{P}_\alpha <\alpha\mid\Omega\mid\alpha>,$$

(44)

which returns the original ensemble average expression.

---

[6] The trace Tr is defined as the sum of the diagonal matrix elements of an operator, where a CON basis is used.

3.1 Properties of the Density Matrix

We have defined the density operator by a sum involving state labels α for the special case of a spin $1/2$ system. The definition

$$\rho = \sum_\alpha \; | \, \alpha ><\alpha \, | \, \mathbf{P}_\alpha \tag{45}$$

is however a general one, if we interpret α as the label for the possible characteristics of a state. Several important general properties of a density operator can now be delineated. The density matrix is Hermitian, hence its eigenvalues are real. The density matrix is also positive definite, which means that all of its eigenvalues are greater or equal to zero. This, together with the fact that the density matrix has unit trace, ensures that the eigenvalues are in the range $[0,1]$.

To prove that the density matrix is positive definite, consider a basis $| \, \nu >$ which diagonalizes the density operator so that

$$
\begin{aligned}
<\nu \, | \, \rho \, | \, \nu > &= \mu_\nu \\
&= \sum_\alpha \mathbf{P}_\alpha <\nu \, | \, \alpha ><\alpha \, | \, \nu > = \sum_\alpha \mathbf{P}_\alpha \; |<\nu \, | \, \alpha >|^2 \geq 0.
\end{aligned}
\tag{46}
$$

Here μ_ν is the νth eigenvalue of ρ and both parts of the final sum above are positive quantities. Hence all of the eigenvalues of the density matrix are ≥ 0 and the density matrix is thus positive definite. If one of the eigenvalues is one, all the others are zero.

Another general property of the density matrix involves the special case of a pure state. If every member of the ensemble has the same quantum state, then only one α (call it α_0) appears and the density operator becomes $\rho = | \, \alpha_0 >< \alpha_0 \, |$. The state $| \, \alpha_0 >$ is normalized to one and hence for a pure state $\rho^2 = \rho$. Using a basis that diagonalizes ρ, this result tells us that the eigenvalues satisfy $\mu_\nu(\mu_\nu - 1) = 0$ and hence for a pure state one density matrix eigenvalues is 1, with all others zero.

In general, an ensemble does not have all of its members in the same state, but has a mixture of possibilities as reflected in the probability distribution $\mathbf{P}_\alpha$. In general, as we show below, we have

$$\rho^2 \leq \rho, \tag{47}$$

with the equal sign holding for pure states. A simple way to understand this relationship is seen by transforming the density matrix to diagonal form, using its eigenstates to form a unitary matrix U_ρ. We have $U_\rho \rho U_\rho^\dagger = \rho_D$, where ρ_D is diagonal using the eigenstates of ρ as the basis, e.g. $<\nu \, | \, \rho_D \, | \, \nu' >= \mu_\nu \delta_{\nu\nu'}$.

20

Here μ_ν again denotes the νth eigenvalue of ρ. We already know that the sum of all these eigenvalue equals 1, that they are real and positive. Since every eigenvalue is limited to be less than or equal to 1, we have $\mu_\nu^2 \leq \mu_\nu$, for all ν. Transforming that back to the original density matrix yields the result $\rho^2 \leq \rho$. Taking the trace of this result yields another test for the purity of the state $\mathrm{Tr}[\rho^2] \leq \mathrm{Tr}[\rho] = 1$. Examples of how to use this measure of purity will be discussed later.

3.1.1 Entropy and Fidelity

As an indication of the rich variety of functionals of ρ that can be defined, let us examine the Von Neumann entropy and the fidelity.

The Von Neumann entropy [14], $S[\rho] = -\mathrm{Tr}[\rho\ \log_2 \rho]$, is a measure of the degree of disorder in the ensemble. Its basic properties are: $S[\rho] = 0$ if ρ is a pure state, and $S[\rho] = 1$ for completely disordered states. See later for an application to the Bell, GHZ , & Werner states and also the *Tutorial.nb* notebook for simple illustrative examples.

It is often of interest to compare two different density matrices that are alternate descriptions of an ensemble of quantum systems. One simple measure of such differences is the fidelity. Consider for example, two pure states

$$\rho = \mid \psi ><\psi \mid \qquad \rho = \mid \widetilde{\psi} ><\widetilde{\psi} \mid, \tag{48}$$

and the associated overlap obtained by a trace method

$$\mathrm{Tr}[\ \rho\ \widetilde{\rho}\ ] = \mathrm{Tr}[\ \mid \psi ><\psi \mid\ \widetilde{\psi} ><\widetilde{\psi} \mid\ ] = \mid <\psi \mid \widetilde{\psi} > \mid^2. \tag{49}$$

Clearly this overlap equals one if the states and associated density matrices are the same and thus serves as a measure to compare states. This procedure is generalized and applied to general density matrices. It is also written in a symmetric manner, with the general definition of fidelity being

$$F[\rho, \widetilde{\rho}] = \mathrm{Tr}[\sqrt{\sqrt{\widetilde{\rho}}\ \rho\ \sqrt{\widetilde{\rho}}}\ ]\ , \tag{50}$$

which has the property of reducing to $\mathrm{Tr}[\rho] = 1$, for $\widetilde{\rho} = \rho$. It also yields $\mid <\psi \mid \widetilde{\psi} > \mid$ in the pure state limit.

3.1.2 Usage

QDENSITY includes commands that produce the Purity and Entropy for a stipulated density matrix ρ, Purity$[\rho]$, Entropy$[\rho]$, and the Fidelity of one specified density matrix relative to another Fidelity$[\rho_1, \rho_2]$.

3.1.3 Composite Systems and Partial Trace

For a composite system, such as colliding beams, or an ensemble of quantum systems each of which is prepared with a probability distribution, the definition of a density matrix can be generalized to a product Hilbert space form involving systems of type A or B

$$\rho_{AB} \equiv \sum_{\alpha,\beta} \mathbf{P}_{\alpha,\beta} \mid \alpha\beta >< \alpha\beta \mid, \tag{51}$$

where $\mathbf{P}_{\alpha,\beta}$ is the joint probability for finding the two systems with the attributes labelled by α and β. For example, α could designate the possible directions $\hat{n}$ of one spin-1/2 system, while β labels the possible spin directions of another spin 1/2 system. One can always ask about the state of system A or B by summing over or tracing out the other system. For example the density matrix of system A is picked out of the general definition above by the following trace steps

$$
\begin{aligned}
\rho_A &= \mathrm{Tr}_B[\rho_{AB}] \\
&= \sum_{\alpha,\beta} \mathbf{P}_{\alpha,\beta} \mid \alpha >< \alpha \mid \mathrm{Tr}_B[\mid \beta >< \beta \mid] \\
&= \sum_{\alpha} (\sum_{\beta} \mathbf{P}_{\alpha,\beta}) \mid \alpha >< \alpha \mid \\
&= \sum_{\alpha} \mathbf{P}_{\alpha} \mid \alpha >< \alpha \mid .
\end{aligned}
\tag{52}
$$

Here we use the product space $\mid \alpha\beta > \mapsto \mid \alpha > \mid \beta >$ and we define the probability for finding system A in situation α by

$$\mathbf{P}_{\alpha} = \sum_{\beta} \mathbf{P}_{\alpha,\beta}. \tag{53}$$

This is a standard way to get an individual probability from a joint probability.

It is easy to show that all of the other properties of a density matrix still hold true for a composite system case. It has unit trace, it is Hermitian with real eigenvalues, etc.

See later for application of these general properties to multi-qubit systems.

3.2 Comments about the Density Matrix

3.2.1 Alternate Views of the Density Matrix

In the prior discussion, the view was taken that the density matrix implements a classical average over an ensemble of many quantum systems, each member

of which interferes quantum mechanically only with itself. Another viewpoint, which is equally valid, is that a single quantum system is prepared, but the preparation of this single system is not pinned down. Instead all we know is that it is prepared in any one of the states labelled again by a generic state label α with a probability $\mathbf{P}_\alpha$. Despite the change in interpretation, or rather in application to a different situation, all of the properties and expressions presented for the ensemble average hold true; only the meaning of the probability is altered.

Another important point concerning the density matrix is that the ensemble average (or the average expected result for a single system prepared as described in the previous paragraph) can be used to obtain these averages for all observables Ω. Hence in a sense the density matrix describes a system and the system's accessible observable quantities. It represents then an honest statement of what we can really know about a system. On the other hand, in Quantum Mechanics it is the wave function that tells all about a system. Clearly, since a density matrix is constructed as a weighted average over bilinear products of wave functions, the density matrix has less detailed information about a system that is contained in its wave function. Explicit examples of these general remarks will be given later.

To some authors the fact that the density matrix has less content than the system's wave function, causes them to avoid use of the density matrix. Others find the density matrix description of accessible information as appealing.

3.2.2   Classical Correlations and Entanglement

The density matrix for composite systems can take many forms depending on how the systems are prepared. For example, if distinct systems A & B are independently produced and observed independently, then the density matrix is of product form $\rho_{AB} \mapsto \rho_A \otimes \rho_B$, and the observables are also of product form $\Omega_{AB} \mapsto \Omega_A \otimes \Omega_B$. For such an uncorrelated situation, the ensemble average factors

$$< \Omega_{AB} > = \frac{\mathrm{Tr}[\rho_{AB}\Omega_{AB}]}{\mathrm{Tr}[\rho_{AB}]} = \frac{\mathrm{Tr}[\rho_A\Omega_A]}{\mathrm{Tr}[\rho_A]}\frac{\mathrm{Tr}[\rho_B\Omega_B]}{\mathrm{Tr}[\rho_B]} \tag{54}$$

as is expected for two separate uncorrelated experiments. This can also be expressed as having the joint probability factor $\mathbf{P}_{\alpha,\beta} \mapsto \mathbf{P}_\alpha\mathbf{P}_\beta$ the usual probability rule for uncorrelated systems.

Another possibility for the two systems is that they are prepared in a coordinated manner, with each possible situation assigned a probability based on the correlated preparation technique. For example, consider two colliding beams, A & B, made up of particles with the same spin. Assume the particles are produced in matched pairs with common spin direction $\hat{n}$. Also assume that the preparation of that pair in that shared direction is produced by design with a

classical probability distribution $\mathbf{P}_{\hat{n}}$. Each pair has a density matrix $\rho_{\hat{n}} \otimes \rho_{\hat{n}}$ since they are produced separately, but their spin directions are correlated classically. The density matrix for this situation is then

$$\rho_{AB} = \sum_{\hat{n}} \mathbf{P}_{\hat{n}} \; \rho_{\hat{n}} \otimes \rho_{\hat{n}}. \tag{55}$$

This is a "mixed state" which represents classically correlated preparation and hence any density matrix that takes on the above form can be reproduced by a setup using classically correlated preparations and does *not* represent the essence of Quantum Mechanics an entangled state.

An entangle quantum state is described by a density matrix (or by its corresponding state vectors) that is not and can not be transformed into the two classical forms above; namely, cast into a product or a mixed form. For example, a Bell state $\frac{1}{2}(\mid 01 > + \mid 10 >)$ has a density matrix

$$\rho = \frac{1}{2}(\mid 01 >< 01 \mid + \mid 01 >< 10 \mid + \mid 10 >< 01 \mid + \mid 10 >< 10 \mid) \tag{56}$$

that is not of simple product or mixed form. It is the prime example of an entangled state.

The basic idea of decoherence can be described by considering the above case with time dependent coefficients

$$\rho = \frac{1}{2}(a_1(t) \mid 01 >< 01 \mid + a_2(t) \mid 01 >< 10 \mid + a_2^*(t) \mid 10 >< 01 \mid + a_3(t) \mid 10 >< 10 \mid). \tag{57}$$

If the off-diagonal terms $a_2(t)$ vanish, by attenuation and/or via time averaging, then the above density matrix does reduce to the mixed or classical form, which is an illustration of how decoherence leads to a classical state.

4 MULTI -QUBIT SYSTEMS

The previous discussion which focused on describing a single qubit, can now be generalized to multiple qubits. Consider the product space of two qubits both in the up state and denote that product state as $\mid 0 \; 0 >= \mid 0 > \mid 0 >$, which clearly generalizes to

$$\mid q_1 \; q_2 >= \mid q_1 > \mid q_2 >, \tag{58}$$

where q_1, q_2 take on the values 0 and 1. This product is called a tensor product and is symbolized as

$$\mid q_1 \; q_2 >= \mid q_1 > \otimes \mid q_2 >, \tag{59}$$

In[4]:= **KetV[{0, 1, 1}]**

In[5]:= **ket123 = ((KetX[0] ⊗ Ket[1]) ⊗ KetX[1])**

In[4]:= **ket12 = KetY[0] ⊗ Ket[1]**

$$
Out[5]= \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \\ -\frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}
\qquad
Out[4]= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

$$
Out[4]= \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{i}{\sqrt{2}} \end{pmatrix}
$$
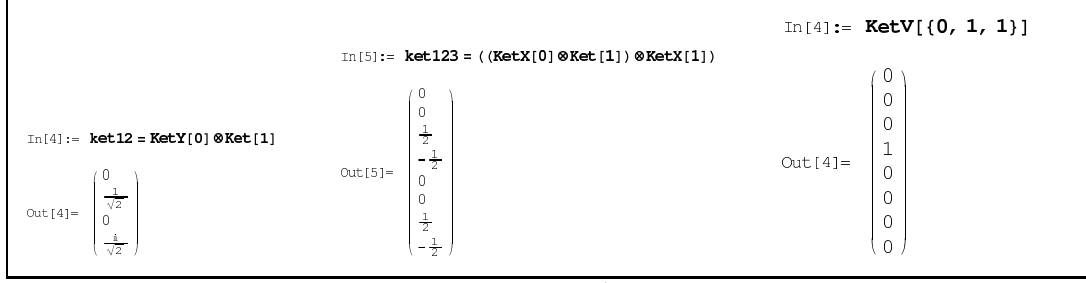
Fig. 2. Simple examples of tensor products of two and three kets.

which generalizes to n_q qubits

$$
\mid q_1\ q_2\ \cdots\ q_{n_q} >= (\mid q_1 > \otimes \mid q_2 >)(\cdots \otimes \mid q_{n_q} >). \tag{60}
$$

In QDENSITY, the kets $\mid 0 >, \mid 1 >$ are invoked by the commands Ket[0] and Ket[1], as shown in Fig. 2, along with the kets $\mid \pm x >$, and $\mid \pm y >$.

Also shown in that figure are the results for forming the tensor products of the kets for two and three qubits, as described next.

4.1 Multi-Qubit Operators

One can also build operators that act in the multi-qubit spin space described above. Instead of a single operator, we have a set of separate Pauli operators acting in each qubit space. They commute because they refer to separate, distinct quantum systems. Hence, we can form the tensor product of the n_q spin operators which for two qubits has the following structure

$$
< a_1 \mid \sigma_i \mid b_1 >< a_2 \mid \sigma_j \mid b_2 >=< a_1 a_2 \mid \sigma_i^{(1)} \sigma_j^{(2)} \mid b_1 b_2 >
$$
$$
=< a_1 a_2 \mid \sigma_i^{(1)} \otimes \sigma_j^{(2)} \mid b_1 b_2 >, \tag{61}
$$

which defines what we mean by the tensor product $\sigma_i^{(1)} \otimes \sigma_j^{(2)}$ for two qubits. The generalization is immediate

$$
(\sigma_i^{(1)} \otimes \sigma_j^{(2)}) \otimes (\sigma_k^{(3)} \otimes \sigma_l^{(4)}) \cdots . \tag{62}
$$

The corresponding steps in QDENSITY are shown in Fig. 3,

For large numbers of qubits a recursive method has been developed (see, *Qdensity.m* and *Tutorial.nb*), which involves specifying the "Length" L= n_q of the qubit array and an array of length L that specifies the Pauli components $i, j, k, l, \cdots$. For example, if $i = 1, j = 0$ there is a σ_x in qubit 1 space and a unit operator σ_0 acting in qubit 2 space. The multi-qubit spin operator is

```
      Constructing Multiqubit Operators

In[2]:= Op1 = ((σ₃ ⊗ σ₂) ⊗ σ₀)

            ⎛ 0   0  -i   0    0   0   0  0 ⎞
            ⎜ 0   0   0  -i    0   0   0  0 ⎟
            ⎜ i   0   0   0    0   0   0  0 ⎟
            ⎜ 0   i   0   0    0   0   0  0 ⎟
Out[2]=     ⎜ 0   0   0   0    0   0   i  0 ⎟
            ⎜ 0   0   0   0    0   0   0  i ⎟
            ⎜ 0   0   0   0   -i   0   0  0 ⎟
            ⎝ 0   0   0   0    0  -i   0  0 ⎠


In[3]:= Op2 = SP[3, {3, 2, 0}]

            ⎛ 0   0  -i   0   0   0   0  0 ⎞
            ⎜ 0   0   0  -i   0   0   0  0 ⎟
            ⎜ i   0   0   0   0   0   0  0 ⎟
            ⎜ 0   i   0   0   0   0   0  0 ⎟
Out[3]=     ⎜ 0   0   0   0   0   0   i  0 ⎟
            ⎜ 0   0   0   0   0   0   0  i ⎟
            ⎜ 0   0   0   0  -i   0   0  0 ⎟
            ⎝ 0   0   0   0   0  -i   0  0 ⎠
```

Fig. 3. Multi-qubit operators in QDENSITY

called $SP[L, \{i, j, k, l, \cdots\}]$. Examples in Fig. 3 include operator tensor products generated directly using the $\otimes$ notation.

4.1.1 Usage

QDENSITY includes a multiqubit spin operator $\mathbf{SP[L, \{}a_1, a_2, .., a_L\mathbf{\}]}$ built from L Pauli spin operators of components $a_1, a_2, .., a_L$. A sample construction is:

$$\text{In}[1] := \mathbf{SP[2, \{2, 3\}]}$$

$$\text{Out}[1] := \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \end{pmatrix}$$

which is equivalent to the tensor product

$$\text{In}[2] := \sigma_\mathbf{2} \otimes \sigma_\mathbf{3}$$

$$\text{Out}[2] := \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \end{pmatrix}$$

The advantage of this command is that it can readily construct large space tensor products.

4.2 General Multi-Qubit Operators

The production of n_q spin space operators provides a complete basis for expressing any operator. This remark is similar to, and indeed equivalent to, the statement that the n_q ket product space is a CON basis. With that remark, we can expand any n_q operator as

$$\Omega = \sum_{\mathbf{a}} \mathcal{C}_{\mathbf{a}} \; \sigma_{a_1}^{(1)} \otimes \sigma_{a_2}^{(2)} \otimes \sigma_{a_3}^{(3)} \cdots \sigma_{a_{n_q}}^{(n_q)} = \sum_{\mathbf{a}} \mathcal{C}_{\mathbf{a}} \; \mathrm{SP}[n_q, \mathbf{a}] \tag{63}$$

where the sum is over all possible values of the array $\mathbf{a} : \{a_1, a_2, a_3, \cdots, a_{n_q}\}$. Here, the multi-qubit spin operator is denoted by $\mathrm{SP}[n_q, \mathbf{a}]$, which is the notation used in QDENSITY. The coefficient $\mathcal{C}_{\mathbf{a}}$ can be evaluated for any given Ω from the overall trace

$$\mathcal{C}_{\mathbf{a}} = \frac{1}{2^{n_q}} \mathrm{Tr}[\, \Omega \; . \mathrm{SP}[n_q, \mathbf{a}]\,]. \tag{64}$$

Because of the efficacy of Mathematica 5.2, the total trace can be evaluated rapidly. This set of coefficients characterizes the operator Ω.

4.2.1 Partial Traces

The advantage of expanding a general operator in the Pauli operator basis is that partial traces can now be generated by manipulating the above coefficients. A partial trace involves tracing out parts of a system; for example, consider the partial trace over qubit two for a three qubit operator

$$\mathrm{Tr}_2[\sigma_i^{(1)} \otimes \sigma_j^{(2)} \otimes \sigma_k^{(3)}] = 2\delta_{j0} \; \sigma_i^{(1)} \otimes \sigma_k^{(3)} \; . \tag{65}$$

Recall that $\mathrm{Tr}[\sigma_i]$ is zero unless we have the unit matrix σ_0 in which case the trace is two. Of course, one could trace out systems 2 and also 1, and then

$$\mathrm{Tr}_{12}[\sigma_i^{(1)} \otimes \sigma_j^{(2)} \otimes \sigma_k^{(3)}] = 2\delta_{i0} \; 2\delta_{j0} \; \sigma_k^{(3)} \; . \tag{66}$$

The subscript on the Tr symbol indicates which qubit operators are being traced out. Note in this case the number of qubits in the result is reduced to $n_q - 2$, where 2 is the length of the subscript array in Tr_{12}. Clearly, the trace reduces the rank [7] of the operator by the number of qubits traced out.

Now we can apply these simple ideas to construct the partial trace of a general operator Ω. Using the linearity of the trace

$$\mathrm{Tr}_{\mathbf{t}}[\Omega] = \sum_{\mathbf{a}} \mathcal{C}_{\mathbf{a}} \; \mathrm{Tr}_{\mathbf{t}}[\sigma_{a_1}^{(1)} \otimes \sigma_{a_2}^{(2)} \otimes \sigma_{a_3}^{(3)} \cdots \; \sigma_{a_{n_q}}^{(n_q)}], \tag{67}$$

---

[7] The rank is the number of qubits n_q.

```
In[7]:= ρ₁₂₃ = ((PX[0]⊗PX[1])⊗PY[0])
        ρ₁ = PTr[{2, 3}, ρ₁₂₃]
        ρ₁₂ = PTr[{1}, ρ₁₂₃]
```

$$\text{Out[7]}= \begin{pmatrix}
\frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} \\
\frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} \\
-\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} \\
-\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} \\
\frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} \\
\frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} \\
-\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} \\
-\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8} & -\frac{i}{8} & -\frac{1}{8} & \frac{i}{8} & \frac{1}{8}
\end{pmatrix}$$

$$\text{Out[8]}= \begin{pmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2}
\end{pmatrix}$$

$$\text{Out[9]}= \begin{pmatrix}
\frac{1}{4} & -\frac{i}{4} & -\frac{1}{4} & \frac{i}{4} \\
\frac{i}{4} & \frac{1}{4} & -\frac{i}{4} & -\frac{1}{4} \\
-\frac{1}{4} & \frac{i}{4} & \frac{1}{4} & -\frac{i}{4} \\
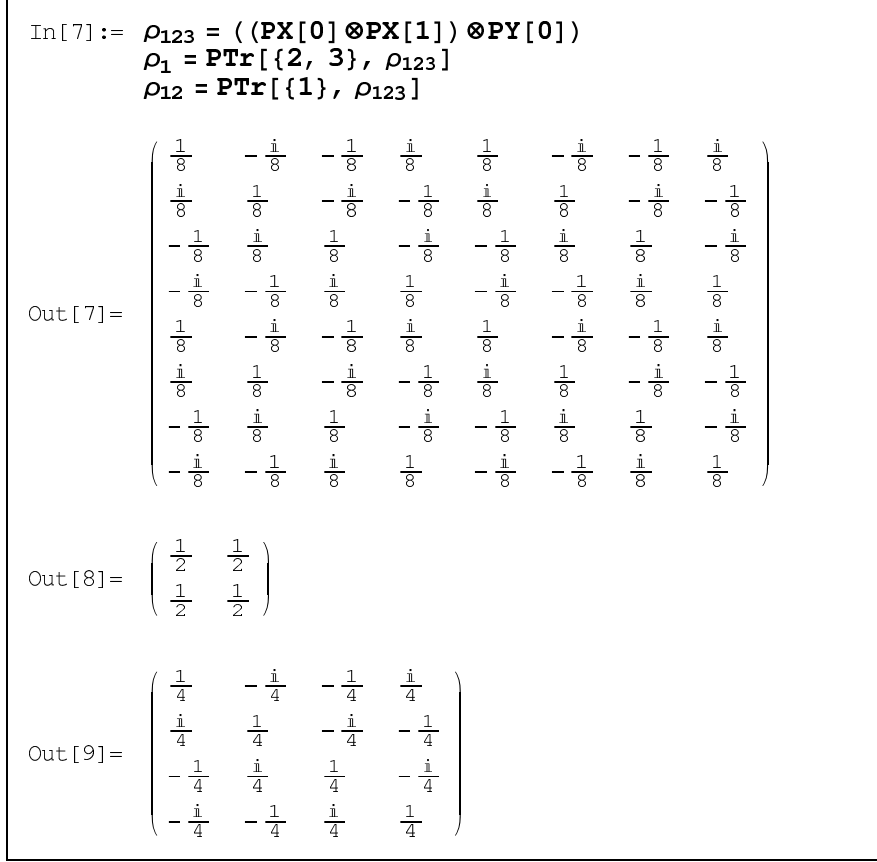-\frac{i}{4} & -\frac{1}{4} & \frac{i}{4} & \frac{1}{4}
\end{pmatrix}$$

Fig. 4. Taking partial traces with QDENSITY

where the array $\mathbf{t} : \{t_1 t_2 \cdots\}$ indicates only those qubits that are to be traced out. For example, $\mathbf{t} : \{25\}$ indicates that only qubits 2 and 5 are traced out.

The procedure for taking a partial trace of a general operator is to determine the total coefficient $\mathcal{C}_\mathbf{a}$ for all of the array $\mathbf{a} : \{a_1 a_2 \cdots a_{n_q}\}$, except for the entries corresponding to the traced out qubit for which we need only the $a_j = 0$ part if say we trace out the jth qubit. From the resultant coefficients, we obtained a reduced set of coefficients, reduced by the number of trace outs. That reduced coefficient is then used to construct the reduced space operator, with a multiplier of 2 included for each traced out qubit. This expansion, reduction, reconstruction procedure might seem complicated, but it has been implemented very efficiently using the power of Mathematica 5.2. See *Qdensity.m* for the explicit construction procedure (which is rather compact). The command used in QDENSITY is PTr[$\mathbf{t}$, Ω] where the trace out of the general operator Ω is specified by the array $\mathbf{t}$. Examples of the partial traces are in Fig. 4.

4.2.2 Usage

QDENSITY includes several commands for taking partial traces. One is $\mathbf{PTr}[\{q_1, q_2, ..., q_M\}, \Omega]$,

28

```
In[10]:= PTr[{1, 2, 3}, SP[5, {1, 3, 3, 0, 1}]]

          ⎛ 0   0   0   0 ⎞
          ⎜ 0   0   0   0 ⎟
Out[10]=  ⎜ 0   0   0   0 ⎟
          ⎝ 0   0   0   0 ⎠


In[11]:= PTr[{3, 4, 5}, SP[5, {1, 3, 0, 0, 0}]]

          ⎛ 0    0   8   0 ⎞
          ⎜ 0    0   0  -8 ⎟
Out[11]=  ⎜ 8    0   0   0 ⎟
          ⎝ 0   -8   0   0 ⎠
```

Fig. 5. Partial Traces of multi-qubit Pauli operators

where the array $q_1, q_2, ..., q_M$ stipulates the space to be traced out. See *Tutorial.nb* and Fig. 4 for examples of these commands.

4.3   Multi-Qubit Density Matrix

The multi-qubit density matrix is our prime example of an operator that we examine in various ways, including taking partial traces. Just as in the prior discussion, a general density matrix can be expanded in a Pauli spin operator basis

$$\rho = \sum_{\mathbf{a}} \mathcal{C}_{\rho_{\mathbf{a}}} \ \mathrm{SP}[n_q, \mathbf{a}], \tag{68}$$

where the coefficient $\mathcal{C}_{\rho_{\mathbf{a}}}$ is real since the density matrix and the Pauli spin tensor product $\mathrm{SP}[n_q, \mathbf{a}]$ are Hermitian. Taking a partial trace follows the rules discussed earlier. Examples are presented in Figs. 5 and 6.

In these examples, we give the case of three qubits reduced to two and then to one. The general expansion for these cases takes on a simple and physically meaningful form and therefore are worth examining. For one qubit, the above expansion is of the traditional form

$$\rho_1 = \frac{1}{2}[\ \mathbf{1} + \vec{P}_1 \cdot \vec{\sigma}], \tag{69}$$

which involves the three numbers contained in the vector $\vec{P}_1$, also know as the polarization of the ensemble. A 2×2 Hermitian density matrix has 4 variables, which is reduced by one by the $\mathrm{Tr}[\rho_1] = 1$ normalization. Thus the polarization vector is a complete parametrization of a single qubit. For a pure state, the magnitude of the polarization vector is one; whereas, the general constraint $\rho^2 \leq \rho$ implies that $\mid P_1 \mid \leq 1$. A graph of that vector thus lies within a unit circle called the Bloch sphere. The physical meaning of $\vec{P}_1$ is that it is the average polarization of an ensemble, which is made clear by forming the

ensemble average of the Pauli spin vector:

$$< \vec{\sigma} >= \frac{\text{Tr}[\rho_1 \vec{\sigma}]}{\text{Tr}[\rho_1]} \equiv \vec{P}_1. \tag{70}$$

Now consider two qubits. The Pauli basis is $\sigma_i \otimes \sigma_j$, and hence the two qubit density matrix has the form

$$\rho_{12} = \frac{1}{4} [\ \mathbf{1} + \vec{P}_1 \cdot \vec{\sigma}_1 \otimes \mathbf{1} + \mathbf{1} \otimes \vec{\sigma}_2 \cdot \vec{P}_2 + \sigma_{1i} \otimes \sigma_{2j} T_{i,j}]$$
$$= \frac{1}{4} [\ \mathbf{1} + \vec{P}_1 \cdot \vec{\sigma}_1 + \vec{P}_2 \cdot \vec{\sigma}_2 + \vec{\sigma}_1 \cdot \overleftrightarrow{T} \cdot \vec{\sigma}_{2j}].$$

This involves two polarization vectors, plus one 3×3 tensor polarization $\overleftrightarrow{T}$ [8] which comes to 15 parameters as indeed is the correct number for a two qubit system $2^2 \times 2^2 - 1$ [9]. The physical meaning is again an ensemble average polarization vector for each qubit system, plus an ensemble average spin correlation tensor

$$< \vec{\sigma}_1 > = \frac{\text{Tr}[\rho_{12} \ \vec{\sigma}_1 \otimes \mathbf{1}_2]}{\text{Tr}[\rho_{12}]} \equiv \vec{P}_1,$$
$$< \vec{\sigma}_2 > = \frac{\text{Tr}[\rho_{12} \ \mathbf{1}_1 \otimes \vec{\sigma}_2]}{\text{Tr}[\rho_{12}]} \equiv \vec{P}_2,$$
$$< \sigma_{1i}\sigma_{2j} > = \frac{\text{Tr}[\rho_{12} \ \sigma_{1i} \otimes \sigma_{2j}]}{\text{Tr}[\rho_{12}]} \equiv T_{ij}. \tag{71}$$

To illustrate a partial trace, consider the trace over qubit 2 of the two qubit density matrix

$$\text{Tr}_2[\rho_{12}] = \rho_1 = \frac{1}{2}[\mathbf{1} + \vec{P}_1 \cdot \vec{\sigma}_1], \tag{72}$$

where we see that a proper reduction to the single qubit space results. Examples of the density matrix for the Bell states and their partial trace reduction to the single qubit operator are presented in Fig. 6.

---

[8] In the tensor term the sum extends only over the $i, j = 1, 2, 3$ components
[9] We see that the number of parameters in a n_q qubit density matrix is thus $2^{n_q} \times 2^{n_q} - 1 = 2^{2n_q} - 1$.

```
In[2]:= Bellstate = (Ket[0] ⊗ Ket[0] + Ket[1] ⊗ Ket[1]) / Sqrt[2]
        ρBell = Bellstate ⊗ Adj[Bellstate]
        Entropy[ρBell]
```

$$\text{Out[2]}= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\text{Out[3]}= \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

Out[4]= 0

The Bell state has entropy 0, is a pure state.

```
In[5]:= ρB1 = PTr[{2}, ρBell]
        ρB2 = PTr[{1}, ρBell]
        Entropy[ρB1]
        Entropy[ρB2]
```

$$\text{Out[5]}= \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

$$\text{Out[6]}= \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

Out[7]= 1

Out[8]= 1

...but, the entropy of each of the subsystems is 1, they are in a completely disordered state!

Fig. 6. Example from *Tutorial.nb*.

4.4 Multi-Qubit States

The procedure for building multi-qubit states follows a path similar to our discussion of operators. First we build the computational basis states, which are eigenstates of the operator $\sigma_z^{(1)} \otimes \sigma_z^{(2)} \otimes \cdots \sigma_z^{(n_q)}$. These states are specified by an array $\mathbf{a} : \{a_1, a_2 \cdots a_{n_q}\}$ of length n_q, where the entries are either one or zero. That collection of binary bits corresponds to a decimal number according to the usual rule $a_1 a_2 \cdots a_{n_q} \rightarrow a_1 2^{n_q} + a_2 2^{n_q-1} + a_{n_q} 2^0$. The corresponding product state $\mid a_1 a_2 \cdots a_{n_q} > \equiv \mid a_1 > \otimes \mid a_2 > \otimes \cdots \mid a_{n_q} >$ can be constructed using the command KetV[{$a_1, a_2, ..$}]. Any single computational basis state consists of a column vector with all zeros except at the location counting down from the top corresponding to its decimal equivalent. Examples of the construction of multiqubit states in QDENSITY are given in Fig. 2. This capability allows one to use QDENSITY without invoking a density matrix approach, which is often desirable to reduce the space requirements imposed by a full density matrix description.

31

4.4.1 Usage

QDENSITY includes commands for one qubit ket vectors in the computational and in the x- and y-basis **Ket**, **KetX**, **KetY**, and also multiqubit product states using the command **KetV[vec]**. Example of its use is

$$\text{In}[1] := \text{KetV}[\{\mathbf{0}, \mathbf{1}, \mathbf{1}\}]$$

$$\text{Out}[1] := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

which is equivalent to

$$\text{In}[1] := (\text{Ket}[\mathbf{0}] \otimes \text{Ket}[\mathbf{1}]) \otimes \text{Ket}[\mathbf{1}]$$

$$\text{Out}[1] := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

5 CIRCUITS & GATES

Now that we can construct multi-qubit operators and take the partial trace, we are ready to examine the operators that correspond to logical gates for single and multi-qubit circuits. These gates form the basic operations that are part of the circuit model of QC. We will start with one qubit operators in a one

qubit circuit and then go on to one qubit operators acting on selected qubits within a multi-qubit circuit. Then two qubit operators in two and multi-qubit situations will be presented.

5.1 One Qubit Gates

5.1.1 NOT

The basic operation NOT is simply represented by the σ_x matrix since $\sigma_x \mid 0 >=\mid 1 >$ and $\sigma_x \mid 1 >=\mid 0 >$.

5.1.2 The Hadamard

For the Hadamard, we have the simple relation

$$\mathcal{H} = \frac{\sigma_x + \sigma_z}{\sqrt{2}} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{73}$$

which can also be understood as a rotation about the $\hat{\eta} = \frac{\hat{x}+\hat{z}}{\sqrt{2}}$ axis by $\gamma = \pi$ since

$$R = e^{-\frac{\gamma}{2}\vec{\sigma}\cdot\hat{\eta}} = \cos\frac{\pi}{2}\,\sigma_0 - i\sin\frac{\pi}{2}\,\vec{\sigma}\cdot\hat{\eta} \rightarrow -i\frac{\sigma_x + \sigma_z}{\sqrt{2}}\ . \tag{74}$$

The Hadamard plays an important role in QC by generating the qubit state from initial spin up or spin down states, i.e.

$$\mathcal{H} \mid 0 >= \frac{\mid 0 > + \mid 1 >}{\sqrt{2}} \qquad \mathcal{H} \mid 1 >= \frac{\mid 0 > - \mid 1 >}{\sqrt{2}}. \tag{75}$$

Having a Hadamard act in a multi-qubit case, involves operators of the type $\mathcal{H} \otimes \mathbf{1} \otimes \mathcal{H}$, for which Hadamards act on qubits 1 and 3 only. The command for this kind of operator in QDENSITY is $\mathsf{Had}[n_q,\mathsf{Q}]$ where n_q is the total number of qubits and the array $Q : q_1, q_2, ...$ of length n_q indicates which qubit is or is not acted on by a Hadamard. The rule used is if $q_i > 0$, then the ith qubit is acted on by a Hadamard, whereas $q_j = 0$ designates that the jth qubit is acted on by a unit 2×2 operator. For example, $\mathsf{Had}[3,\{1,0,1\}]$ has a Hadamard acting on qubits 1 and 3 and a unit 2×2 acts on qubit 2, which is the case given above. To get a Hadamard acting on all qubits, include all qubits in Q, e.g., use Q={1,1,1,....}. Thus, an operator $\mathsf{HALL}[L]=\mathsf{Had}[L,\{1,1,1....\}]$ is also implemented where the array of 1's has length n_q of all the qubits. Another QDENSITY command $\mathsf{had}[n_q\,,\mathsf{q}]$ is for a Hadamard acting on one qubit q out of the full set of n_q qubits. So QDENSITY facilitates the action of a one qubit operator in a multi-qubit environment.

```
NiceRotation = RotX[θ₁] ⊗ RotY[θ₂]
```

$$\begin{pmatrix} Cos[\frac{\theta_1}{2}]\,Cos[\frac{\theta_2}{2}] & -Cos[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & -\mathrm{i}\,Cos[\frac{\theta_2}{2}]\,Sin[\frac{\theta_1}{2}] & \mathrm{i}\,Sin[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] \\ Cos[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & Cos[\frac{\theta_1}{2}]\,Cos[\frac{\theta_2}{2}] & -\mathrm{i}\,Sin[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & -\mathrm{i}\,Cos[\frac{\theta_2}{2}]\,Sin[\frac{\theta_1}{2}] \\ -\mathrm{i}\,Cos[\frac{\theta_2}{2}]\,Sin[\frac{\theta_1}{2}] & \mathrm{i}\,Sin[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & Cos[\frac{\theta_1}{2}]\,Cos[\frac{\theta_2}{2}] & -Cos[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] \\ -\mathrm{i}\,Sin[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & -\mathrm{i}\,Cos[\frac{\theta_2}{2}]\,Sin[\frac{\theta_1}{2}] & Cos[\frac{\theta_1}{2}]\,Sin[\frac{\theta_2}{2}] & Cos[\frac{\theta_1}{2}]\,Cos[\frac{\theta_2}{2}] \end{pmatrix}$$

Fig. 7. Example of multiqubit rotation using QDENSITY.

5.1.3 Usage

QDENSITY includes several Hadamard commands. For single qubit cases use
either $\mathcal{H}$ or **had[1,1]**. For a Hadamard acting on a single qubit within a set
of L qubits use had[L,q]; for a set of Hadamards acting on selected qubits
use Had[**L, { $0, 1, 0, 1 \cdots$ }],** and for Hadamards acting on all L qubits use
HALL[L]. These are demonstrated in the tutorial.

5.1.4 Rotations

One can use the rotation operator R to produce a state in any direction.
A rotation *about* an axis $\hat{\eta}$ is given in Eq. (25). For special cases, such as
the $\hat{x}, \hat{y}, \hat{z}$ and $\gamma = \pi$, the expanded form reduce to $-i\sigma_x, -i\sigma_y$ and $-i\sigma_z$,
respectively. For a general choice of rotation, one can use the "MatrixExp"
command directly, or use the spinor rotation matrix for rotation *to* angles θ, ϕ.

For a multi-qubit circuit, the rotation operator for say qubits 1 and 3 can
be constructed using the command $R_{\gamma_1} \otimes \mathbf{1} \otimes R_{\gamma_2} \otimes \mathbf{1} \otimes \cdots$, with associated
rotation axes. Examples from QDENSITY are given in Fig. 7.

5.1.5 Usage

Rotation commands for rorarions about the x-, y- or z- axis by an ankle θ are
included in QDENSITY : RotX[θ], RotY[θ], RotZ[θ] In addition, **Rotqbit[v,t]** builds
the matrix corresponding to a rotation around a general axis axis v by an
angle t.

5.2 Two Qubit Gates

To produce a quantum computer, which relies on quantum interference, one
must create entangled states. Thus the basic step of two qubits interacting
must be included. The interaction of two qubits can take many forms de-
pending on the associated underlying dynamics. It is helpful in QC, to isolate
certain classes of interactions that can be used as logical gates within a circuit
model.

5.2.1 CNOT

The most commonly used two-qubit gate is the controlled-not (CNOT) gate. The logic of this gate is summarized by the expression

$$\mathbf{CNOT} \mid c, t >= \mid c, t \oplus c >,$$

where $c = 0, 1$ is the control bit and $t = 0, 1$ is the target bit. In a circuit diagram the $\bullet$ indicates the control qubits and the $\oplus$ indicates the target qubit.
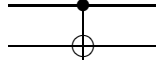
The final state of the target is denoted as "$t \oplus c$" where $\oplus$ addition is understood to be modular base 2. Thus, the gate has the logical role of the following changes (control bit first) $\mid 00 > \mapsto \mid 00 >; \mid 01 > \mapsto \mid 01 > ; \mid 10 > \mapsto \mid 01 >; \mid 11 > \mapsto \mid 10 >$. All of this can be simply stated using projection and spin operators as

$$\mathbf{CNOT}[c, t] = \mid 0 >_c < 0 \mid \otimes \mathbf{I}_t + \mid 1 >_c < 1 \mid \otimes \sigma_x^t, \tag{76}$$

with c and t denoting the control and target qubit spaces. The CNOT, which is briefly expressed as $\mathbf{CNOT} = \mathcal{P}_0 \mathbf{I} + \mathcal{P}_1 \sigma_x$, is called the controlled-not gate since $\mathsf{NOT} \equiv \sigma_x$. A matrix form for this operator acting in a two qubit space is

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{77}$$

The rows & columns are ordered numerically as: $00, 01, 10, 11$.

The CNOT gate is used extensively in QC in a multi-qubit context. Therefore, QDENSITY gives a direct way to embed a two-qubit CNOT into a multi-qubit environment. The command is CNOT $[n_q, c, t]$ where n_q is the total number of qubits and c and t are the control and target qubits respectively as in the following examples: If the number of qubits is 6, and a CNOT acts with qubit 3 as the control and 5 is the target, the operator (which is a $2^6 \times 2^6$ matrix) is invoked by the command $\mathbf{CNOT}[6, 3, 5]$. The command $\mathbf{CNOT}[6, 5, 3]$ has 6 qubits, with qubit 5 the control and 3 the target. The basic case in Eq.(77) is therefore just $\mathbf{CNOT}[2, 1, 2]$.

5.2.2 CPHASE

Other two qubit operators can now be readily generated. The CPHASE gate, which plays an important role in the cluster model of QC is simply a controlled σ_z

$$\mathbf{CPHASE}[c,t] = |\,0>_c< 0\,| \otimes \mathbf{I}_t + |\,1>_c< 1\,| \otimes \sigma_z^t, \tag{78}$$

which in two qubit space has the matrix form

$$\mathbf{CPHASE} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{79}$$

The multi-qubit version is $\mathbf{CPHASE}[n_q, c, t]$, with the same rules as for the **CNOT** gate.

5.2.3 Other Gates

The generation of other gates, such as swap gates [10], Controlled-$i\sigma_y$ (also known as a CROT gate) are now clearly extensions of the prior discussion.

The swap gate swaps the content of two qubits. It can be decomposed in a chain of CNOT gates:

$$|\Psi_1\rangle \quad\quad\quad\quad\quad\quad |\Psi_2\rangle \tag{80}$$
$$|\Psi_2\rangle \quad\quad\quad\quad\quad\quad |\Psi_1\rangle$$

Another example is **CROT**

$$\mathbf{CROT}[c,t] = |\,0>_c< 0\,| \otimes \mathbf{I}_t + |\,1>_c< 1\,| \otimes i\,\sigma_y^t, \tag{81}$$

which, in two qubit space, has the matrix form

$$\mathbf{CROT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}. \tag{82}$$

A multi-qubit version of $\mathbf{CROT}[n_q, c, t]$, with the same rules as for the **CNOT** gate can easily be generated by a modification of *Qdensity.m*.

---

[10] See the QDENSITY command Swap .

Indeed, the general case of a controlled-Ω, where Ω is any one-qubit operator is now clearly

$$C\Omega[c,t] = \mid 0 >_c < 0 \mid \otimes \mathbf{I}_t + \mid 1 >_c < 1 \mid \otimes \Omega^t, \qquad (83)$$

with corresponding extensions to matrix and multi-qubit renditions.

5.2.4 Usage

QDENSITY includes the following two-qubit operators, acting between qubits c(control) and t(target) imbedded in a system of L qubits: CNOT [L,c,t], CPHASE [L,c,t], ControlledX [L,c,t], ControlledY [L,c,t] and Swap [L,q_1,q_2]. A generic two qubit operator within a multiqubit system involving operators $Op1$ and $Op2$ is **TwoOp**[L,q_1,q_2,Op1,Op2].

5.3 Three Qubit Gates

The above procedure can be generalized to three qubit operators. The most important three qubit gate is the Toffoli [1]gate, which has two control bits that determine if a unit or a NOT(σ_x) operator acts on the third (target) bit. The projection operator version of the Toffoli is simply

$$\mathbf{Toffoli} \equiv \mathcal{P}_0 \otimes \mathcal{P}_0 \otimes \mathbf{1} + \mathcal{P}_0 \otimes \mathcal{P}_1 \otimes \mathbf{1} + \mathcal{P}_1 \otimes \mathcal{P}_0 \otimes \mathbf{1} + \mathcal{P}_1 \otimes \mathcal{P}_1 \otimes \sigma_x, \quad (84)$$

which states that the third qubit is flipped only if the first two(control) qubits are both 1.

For a multi-qubit system the QDENSITY command Toffoli $[n_q ,q_1, q_2, q_3]$ returns the Toffoli operator with q_1 and q_2 as control qubits, and q_3 as the target qubit within the full set of n_q qubits.

The Toffoli gate can be specialized or reduced to lower gates and is a universal gate.

5.3.1 Usage

QDENSITY includes a generic three qubit operator within a multiqubit system involving operators $Op1$, $Op2$, and $Op3$: **ThreeOp**[L,q_1,q_2,q_3,Op1,Op2,Op3]. The Toffoli gate is a special case and is invoked by the command Toffoli [L,c,c,t], where c, c, t specifies the two control and the one target qubit out of the set of L qubits.

Let's produce a uniform superposition of 4 qubits:

```
ρᵢ = (((𝒫₀⊗𝒫₀)⊗𝒫₀)⊗𝒫₀);
ρHALL = HALL[4].ρᵢ.Adj[HALL[4]]
```

$$
\begin{pmatrix}
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16}
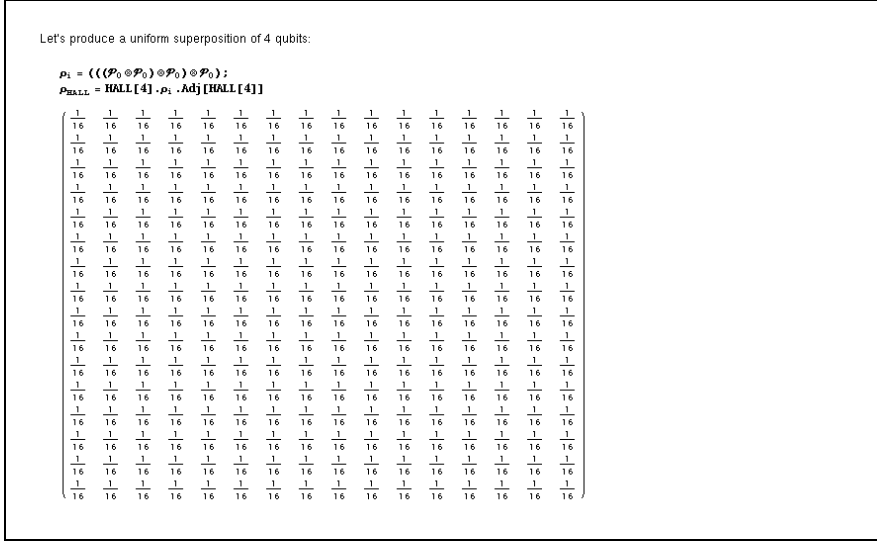\end{pmatrix}
$$

Fig. 8. Construction of a uniform four qubit state.

6 SPECIAL STATES

As a prelude to discussing QC algorithms, it is useful to examine how to produce several states that are part of the initialization of a quantum computer. These are the uniform superposition, the two-qubit Bell, [8] three-qubit GHZ, [9] and Werner [10] states.

6.1 Uniform superposition

In many QC processes, the first step is to produce an initial state of the n_q qubits that is a uniform superposition of all of its possible computational basis states. It is the initialization of this superposition that allows a QC to address lots of questions simultaneously and is often referred to as the "massively parallel" feature of a quantum computer.

The steps start with a state of all spin-up $| 0000 \cdots >$, then every qubit is acted on by a Hadamard $\mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \otimes \ldots$, which is done by the QDENSITY command HALL$[n_q]$. Thus each up state is replaced by $\mathcal{H} | 0 >= \frac{|0>+|1>}{\sqrt{2}}$, and we have the uniform superposition

$$
| \Psi >= \text{HALL}[n_q] | 0 >= \frac{1}{2^{n_q/2}} \sum_{x=0}^{2^{n_q}-1} | x >, \tag{85}
$$

where x is the decimal equivalent to all possible binary numbers of length n_q.

An example of this process, including the associated density matrices, is in Fig. 8.

6.2   Bell States

The singlet and triplet states are familiar in QM as the total spin zero and one states, with zero spin projection ($M = 0$). They are the basic states that enter into the EPR discussion and they are characterized by their "entanglement." Bell introduced two more combinations, so that for two spin 1/2 systems the four Bell states are:

$$| B_{00} >= \frac{1}{\sqrt{2}} \mid 00 > + \mid 11 >$$

$$| B_{01} >= \frac{1}{\sqrt{2}} \mid 01 > + \mid 10 >$$

$$| B_{10} >= \frac{1}{\sqrt{2}} \mid 00 > - \mid 11 >$$

$$| B_{11} >= \frac{1}{\sqrt{2}} \mid 01 > - \mid 10 >, \tag{86}$$

or in one line $\mid B_{ab} >= \frac{1}{\sqrt{2}} \mid 0b > +(-1)^a \mid 1\bar{b} >$, where $\bar{q}$ is the NOT[q] operation.

A circuit that produces these states, starting from the state $\mid ab > (a, b = 1, 0)$ consists of a Hadamard on qubit one, followed by a CNOT . The QDENSITY version is thus:
B [a_, b_] :=CNOT [2,1,2]Had[2,{1,0}](Ket[a]⊗Ket[b]).

The density matrix version involves defining the unitary transformation $U \equiv$ **CNOT**$[2, 1, 2]$.Had$[2, 1, 0]$ and an initial density matrix $\rho_{ab}^I \equiv \mid ab >< ab \mid$, then evolving to the density matrix for each of the Bell states

$$\rho_{ab}^{Bell} = U \cdot \rho_{ab}^I \cdot U^\dagger. \tag{87}$$

In Fig. 9 part of this process taken from *Tutorial.nb* is shown. The tutorial includes a demonstration that the Bell states have zero polarization (as is obvious from their definition), and a simple diagonal form for the associated tensor polarization $\overleftrightarrow{T}$.

Another useful application shown in the tutorial is that taking the partial traces of the Bell state density matrices, yield non pure single qubit density matrices and that the associated von Neumann entropy defined by $S[\rho] = -\text{Tr}[\rho \, \log_2 \rho]$ is zero for the Bell states, but 1 for the single qubit density matrices ρ_1 and ρ_2. Thus each qubit is in a more chaotic state, which physically means they have zero average polarization. [11] This property is an indication of the entanglement of the Bell states.

---

[11] Since many different state vectors can yield a net zero average polarization, it is

The Bell state operator of qubits i and j can be decomposed into the action of a Hadamard on qubit **i** and a CNOT gate on qubits **i** and **j**.

```
In[51]:=  Bellop[qi_, qj_] := CNOT[2, qi, qj].had[2, qi]
```

The density matrix for any of the four Bell states

```
In[52]:=  ρB00  := Bellop[1, 2].ρ00.Adj[Bellop[1, 2]];
          ρB01  := Bellop[1, 2].ρ01.Adj[Bellop[1, 2]];
          ρB10  := Bellop[1, 2].ρ10.Adj[Bellop[1, 2]];
          ρB11  := Bellop[1, 2].ρ11.Adj[Bellop[1, 2]];
```

Checks two Bell states

```
In[56]:=  ρB00
          ρB01
```

$$
Out[56]= \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}
$$

$$
Out[57]= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
$$

Fig. 9. Example from *Teleportation.nb*

6.3   GHZ States

Three qubit states that are similar in spirit to the Bell states, were introduced by Greenberger, M. A. Horne, and A. Zeilinger [9]. The basic GHZ state is

$$
| \Psi >=| \text{GHZ} >= \frac{1}{\sqrt{2}} \left(| 000 > + | 111 > \right), \tag{88}
$$

which may be written:

$$
| GHZ >= U_{GHZ} | 000 > \tag{89}
$$

with $U_{GHZ} = \text{CNOT}[3,1,2].\text{CNOT}[3,1,3].\text{had}[3,1]$ which corresponds to the following circuit:
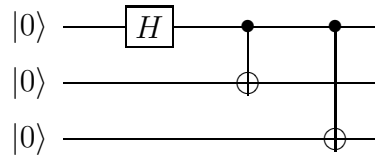
A complete set of eight GHZ states can be produced by the step

$$
U_{\text{GHZ}} | abc >=| \text{GHZ}_{abc} >= \frac{1}{\sqrt{2}}(| 0bc > +(-1)^a | 1\bar{b}\bar{c} >). \tag{90}
$$

---

clear that the density matrix stores less information than in a state vextor, albeit realistic statistical, information.

The **GHZ state** may be prepared by acting on a state with three qubits all initialized to 0 with the following operator:

In[58]:= GHZop = CNOT[3, 1, 3].CNOT[3, 1, 2].had[3, 1];

Then the density matrix for this GHZ state is:

In[59]:= ρGHZ = GHZop.((𝒫₀⊗𝒫₀)⊗𝒫₀).Adj[GHZop]

$$
\text{Out[59]=} \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}
$$

Fig. 10. Construction of a GHZ state. Example from *Tutorial.nb*

For all eight of these three qubit states, the associated density matrix can be formed $\rho_{123}^{abc} = \mid \text{GHZ}_{abc} >< \text{GHZ}_{abc} \mid$ and are seen in *Tutorial.nb* to have a simple structure. Taking partial traces to generate the two qubit $\rho_{12}, \rho_{13}, \rho_{23}$ and single qubit ρ_1, ρ_2, ρ_3 density matrices, we see that for these states every qubit has zero polarization and a simple structure for the pair and three qubit correlation functions. In addition, the entropy of these GHZ set of states is zero and the sub-entropies of the qubit pairs and single qubits are all 1, corresponding to maximum disorder.

A sample GHZ realization in QDENSITY is given in Fig. 10

6.4 Werner States

Another set of states were proposed by Werner [10]. They are defined in terms of a density matrix:

$$\rho_W = \lambda\rho_B + (1-\lambda)\rho_u \otimes \rho_u, \tag{91}$$

where $0 \leq \lambda \leq 1$ is a real parameter. The limit $\lambda = 1$ yields a completely entangled Bell state density matrix $\rho_B = \mid B_{ab} >< B_{ab} \mid$, whereas lower values of λ reduce the entanglement. The $\lambda = 0$ limit gives a simple two qubit product $\rho_u \otimes \rho_u$, where $\rho_u = \frac{1}{2}$ is the density matrix for a single qubit with zero polarization, i.e. it corresponds to a chaotic limit for each qubit. Therefore, the parameter λ can alter the original fully entangled Bell state by introducing chaos or noise. Therefore, the Werner state is called a state of noisy entanglement.

The entropy of a two qubit Werner state as a function of λ ranges from two [12] for $\lambda = 0$, to zero for $\lambda = 1$. The entropy of the single qubit state is 1. See *Tutorial.nb* for a sample Werner QDENSITY realization.

---

[12] This corresponds to an entropy per qubit of 1.

7 TELEPORTATION

To understand QC teleportation, let us first consider classical teleportation, which entails only classical laws. For example, suppose Alice measures a vase using a laser beam to measure its dimensions, shape, color and decoration. She therefore has a full binary description of the vase in 3-D. Bob has all the material to make another vase and, upon receiving the file that Alice sends him by computer, is able to make an exact copy of the vase. There are only local operations (LO) (measuring and sending by Alice and reconstruction by Bob) and a classical communication (CC); so this called a LOCC process. How does this differ from teleportation using Quantum Physics?

In Quantum Mechanics, measurement affects the sample; as a result, after collecting the information to send to Bob, the original sample is no longer in its original state. In the classical case, one ends up with two identical copies. In the QC case, Bob has the only extant system. Another difference is that in the QC case, Alice and Bob share an entangled state, say an EPR or a Bell state, Alice entangles the original system with one member of the pair, then measures and by LOCC sends Bob her result. By virtue of the shared entanglement, information is shared by the LOCC and by the Quantum effect of sharing the entangled state. Some information is transmitted by a "Quantum channel." Therefore, the information sent by computer is less than needed in the classical case, because it is supplemented by the Quantum transfer of entanglement. The strange nature of Quantum transportation is thus no stranger than the EPR/Bell effect, which has been affirmed experimentally.

To understand these general remarks, let us use QDENSITY to examine three cases.

7.1 One Qubit Teleportation

Suppose Alice has one qubit q_1 in an unknown state $\mid \Psi >= a_0 \mid 0 > +a_1 \mid 1 >$, with an associated spin density matrix $\rho_0 \equiv \mid \Psi >< \Psi \mid$. In QDENSITY, such a state is generated randomly. Bob and Alice share a two qubit entangled state, which we take as one of the Bell states B_{q_2,q_3}. This Bell state could be provided by an outside EPR purveyor, but for convenience let us assume that Bob produces the entangled pair and sends one member of the pair $\mid q_2 >$ to Alice, as shown in Fig. 11. Alice then entangles her state $\mid \Psi >$, using the inverse of the steps that Bob employed to produce the entangled pair, and then she measures the state of her $\mid \Psi > \otimes \mid q_2 >$, which yields a single number between zero and three (or one of the binary pairs $00; 01; 10; 11$). Alice transmits by CC that number to Bob, who then knows what to do to his qubit

q_3 in order to slip it over to the state $\mid \Psi >$, without making any measurements on it to avoid its destruction. In the end, Alice no longer has a qubit in the original state, but by LOCC and shared entanglement, she is happy to know that Bob has such a qubit, albeit not made of the original material. The only material transmitted is the single member of the entangled pair.
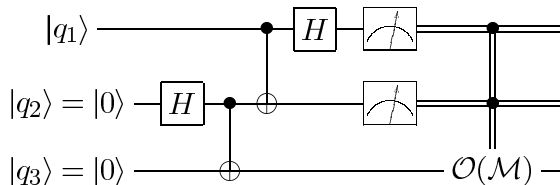

Fig. 11. Schematic circuit for one qubit quantum teleportation.

In QDENSITY , the following detailed steps are shown in the file *Teleportation.nb*: construction of an initial random qubit for Alice, Bob's entanglement process, Alice's entanglement, measurement and CC and Bob's consequent actions. This process is described using the density matrix language. A rendition using the quantum states directly is easily generated.

7.2   Two Qubit Teleportation

7.2.1   Two EPR Teleportation

A similar process can be invoked to teleport an initially unknown state of two qubits $\mid \Psi >= \sum_{i,j=0,1} a_{ij} \mid q_{1i} q_{2j} >$. In the special case that the unknown state is one of the Bell states, it can be transported using the procedure shown in Fig. 12. Bob now prepares a three qubit entangled GHZ state using Hadamards and CNOTs and then sends one of the three qubits q_3 over to Alice, who entangles that one with her original q_1, q_2 qubits $\mid \Psi >_{12} \otimes \mid q_3 >$ and then measures the state of her three qubits, with the result of a number between zero and seven (e.g. one of the binary results $000; 001; 010, 011; 100; 101; 110; 111$). She transmits that decimal number to Bob by CC (a phone call say), who then knows what to do to put his two qubits q_4, q_5 into the original state $\mid \Psi >$. Again all the steps are presented in detail in QDENSITY in the file *Teleportation.nb*.

7.2.2   General Two Qubit Teleportation

The two qubits can be in a more general state than in the above discussion which was restricted to being one of the Bell states. In this case, Bob needs to entangle four qubits by a chain of Hadamard and CNOT gates as shown in Fig. 13. He then sends two qubits q_3, q_4 over to Alice who entangles her two with them. That is, she entangles $\mid \Psi >_{12} \otimes \mid q_3 > \otimes \mid q_4 >$, then
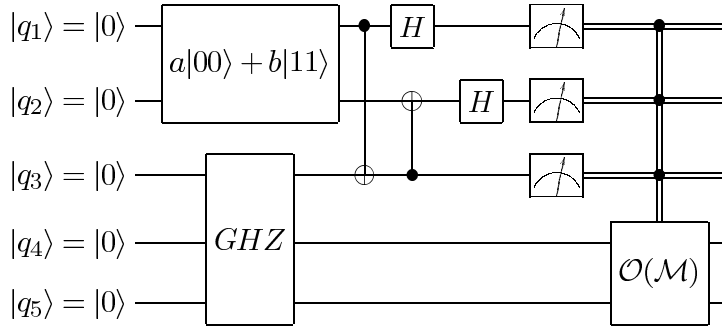
43

Fig. 12. EPR teleportation using a GHZ entangled state.

measures them with a decimal number result that is between zero and 14 or a binary measurement of: $0000; 0001; 0010; 0011; \cdots; 1111$. With that number, Bob knows what to do and places his two qubits into the original state $| \Psi >$. Again see *Teleportation.nb* for the detailed layout.
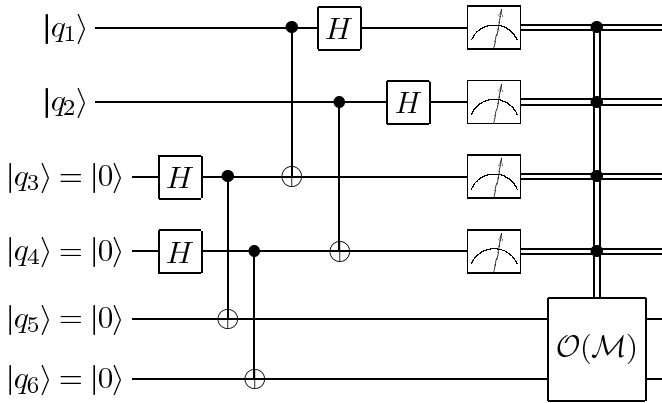

Fig. 13. Schematic circuit for two qubit quantum teleportation.

8 GROVER'S SEARCH

Assume you have a set of items and you want to find a particular one in the set which has been marked beforehand. Let us further restrict the problem by saying that you are only allowed to ask yes/no questions, e.g. "Is this item the item?" In a disordered database with N items and one marked item that problem would require on the order of N trials to find the marked item. Quantum mechanics allows the states of a system to exist in superpositions and thus in many cases permits one to parallelize the process in some sense. Grover [5] proposed an algorithm that lowers the number of trials needed to $O(\sqrt{N})$ by making clever use of interference and superposition. He based his ideas on an analogy to the multiple slit problem [6].

Fig. 14. Example from *Teleportation.nb*

8.1 The Oracle

Grover's search algorithm relies on the use of an Oracle. The idea underlying the Oracle is the following: the Oracle is a function that can recognize a solution to a problem although it may not know how to solve the problem. In that sense a classical example could be a lock and a key, the problem of finding the proper key that would open a lock out of a bunch of keys illustrates the role of the lock as an Oracle: you select a key and you try it, the Oracle would tell you whether that was the correct key or not, but you cannot ask the lock to single out the key from the bunch. The essential difference between a classical and a quantum Oracle is that in the quantum case the Oracle can act on an input which is a superposition of all states. In our example that would mean that we can try a superposition of all the keys at the same time.

The above description of the role of an Oracle takes on the explicit form

$$\mathbf{ORACLE} \mid x >_N \mid y >_1 = \mid x >_N \mid y \oplus f(x) >_1, \tag{92}$$

which involves an N qubit and a single qubit product space and a specified function $f(x)$. The matrix form of the Oracle is

$$< x' \mid < y' \mid \mathbf{ORACLE} \mid x > \mid y >= \delta_{x'x} < y' \mid y \oplus f(x) > . \tag{93}$$

Examples of this Oracle matrix for single and double marked items are given in detail in the Grover.nb notebook, where the "inversion about the mean" process is also presented and explained in detail.

A schematic description of the searching process is:
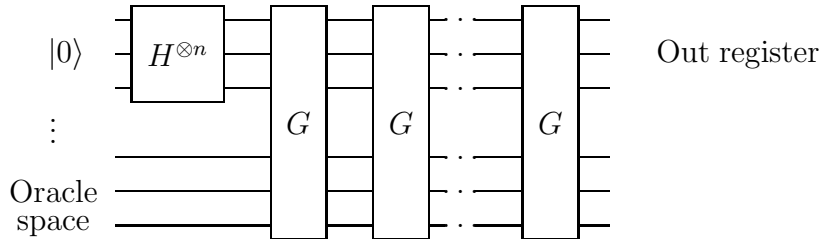
The basic steps, which are detailed in *Grover.nb*, consist of applying a Hadamard to all qubits in the register while setting the Oracle to recognize the marked item. Then we need to construct the Grover operator and apply it a number of times. Finally we measure the output register which then will give the marked item with high probability. The probability of success depends on the number of times the Grover operator has been applied, as can be studied in the notebook.

Grover.nb contains two examples, the first one has only one marked item in the full database. The size of the database, given by the number of qubits, can be varied, together with the number of times the Grover operator is applied.

8.3 Two marked items

The second example includes two marked items in the database (and may be generalized to the case of M marked items). Of course, one needs to enlarge the register and Oracle space.

9 SHOR'S ALGORITHM

Shor's factoring algorithm is the most celebrated quantum algorithm, partly due to its powerful use of quantum superposition and interference to tackle one of the problems upon which most of our secure bank transactions rely. The factoring algorithm, described in detail in the notebook *Shor.nb*, essentially first relates the problem of factoring a number, N, made up of two prime numbers, $N = p \times q$, to the problem of finding the period of a particular function

$x^j \mod (N)$, being x a coprime [13] to N smaller than N. Then finding the period is related to the computation of the Quantum Fourier Transform (QFT) (analog to the discrete Fourier transform), for which a very effective quantum algorithm exists.

Schematically, the procedure is the following: build two registers, 1 and 2, with n_1 and n_2 qubits each, initially set to $|0>$. Then Hadamard the first register, the state of the full system then reads:

$$\Psi = \frac{1}{2^{n_1}} \sum_{i=0}^{2^{n_1}-1} |i>_1 \otimes |0>_2 . \tag{94}$$

Then we take a number x smaller than N and coprime with it and load the second register with the function $x^i \mod (N)$, giving:

$$\Psi = \frac{1}{2^{n_1}} \sum_{i=0}^{2^{n_1}-1} |i>_1 \otimes |x^i \mod N>_2 . \tag{95}$$

At this point, a measurement is performed on the second register and then one applies the QFT to the first register. From that value measured in the first register, one is able, with a certain probability, to factor the original number N.

A detailed study of the performance of the algorithm, e.g. analysis of probabilities of success depending on the size of register considered experimentally can be done within the notebook. A thorough theoretical description of the algorithm can be found in Refs. [7,15].

Shor.nb contains four slightly different approaches to the algorithm, mainly differing in the procedure used to compute the QFT. The most constructive case and also the one appropriate when studying possible noise effects on parts of the circuit, is the density matrix one. There the QFT (see also *QFT.nb*) is obtained using unitary operators in the same way as occurs experimentally. That full QFT treatment implies that the QM method is not practicable for heavy computing simulations, say when number of qubits goes above 10.

Then there are three other cases, two of them using the state vector expression for the QFT with and without explicit construction of both registers.

Finally the last example, makes use of the already implemented discrete Fourier transform in Mathematica. This example is thus useful when emphasis is on

---

[13] Two positive integers a and b are said to be coprime if they have no common factor other than 1, that is, their greatest common divisor is 1.
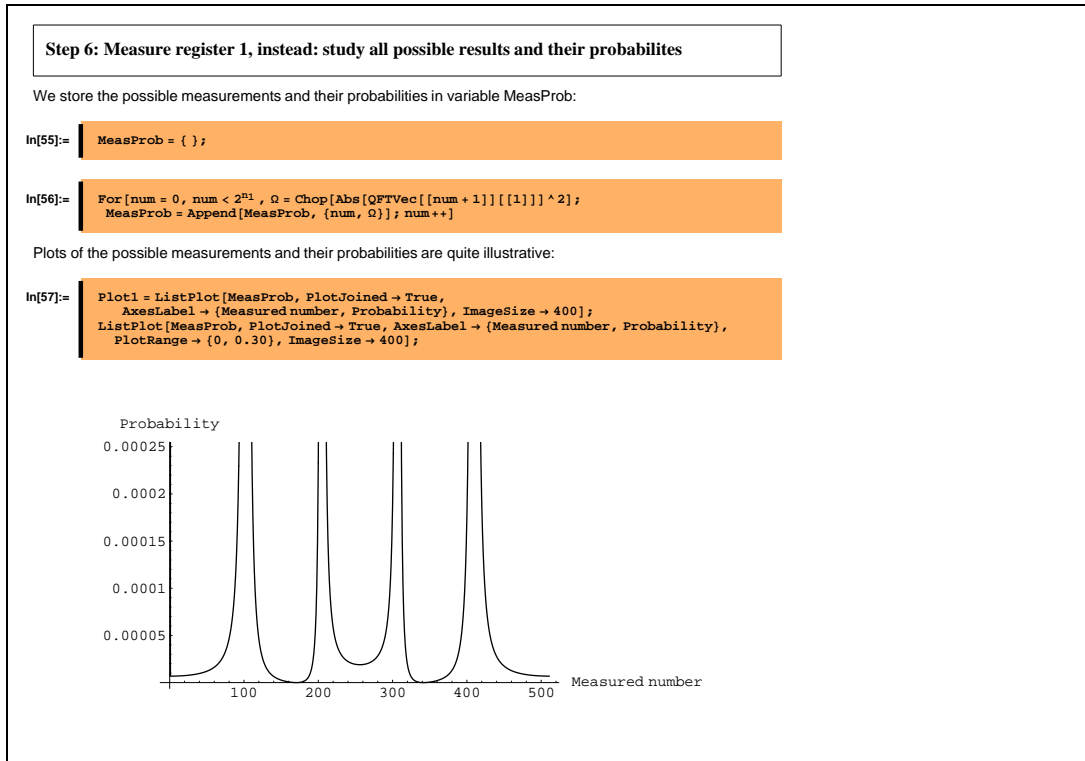
Fig. 15. Example from *Shor.nb*

studying larger numbers to check probabilities of success, without concern for the actual quantum mechanical way of building the QFT. In Fig. 15 a snapshot from *Shor.nb* is shown.

10   CLUSTER MODEL

An alternative to the circuit model for QC has been suggested in a series of recent papers [13]. The basic idea of this approach is to start with an initial state that is highly entangled by virtue of two-qubit CPHASE operations between nearest neighbors in a cluster of qubits. The CPHASE operations could be generated somehow by Ising model spin-spin interactions. Once an appropriate cluster is designed, then a carefully planned set of single qubit measurements are made in various directions. The results of those measurements are passed on by classical communications, until one reaches a final qubit, or set of qubits, from which a result can be deduced once a local correction involving Pauli operators and the binary results of the measurement is invoked. This method is being developed, with the procedures for general algorithms still being formulated. It is however novel and promising, especially since it involves single qubit measurements, can generate gates without use of magnetic fields to rotate spins, and holds forth the promise of error stability. It does require however a large increase in the number of qubits. The fact that
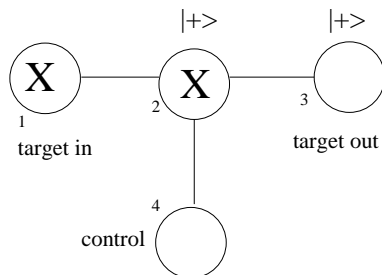
Fig. 16. Cluster model of CNOT gate; see: *Cluster.nb* and [13].

measurement collapse out qubits and essentially destroys the initial state is why this approach is called "one-way computing." Of course, one can reconstruct the initial state and try again, which is typical of QC. One can also reuse qubits after measurement.

It turns out that the modular nature of QDENSITY allows one to use it to demonstrate and test some of the basic ideas of the cluster model of QC. The first illustration is the basic transport process involving two qubits. The two qubits are initially in a $| \Psi > \otimes | + >$ state, where $| \Psi >$ is a general unknown one qubit state. Then a CPHASE operator acts between the two qubits, followed by a measurement of qubit one in the $| +x > \equiv | + >$ direction with a result of either $a =$ zero or one. The second qubit proves to be in the state $\sigma_x^a | \Psi >$. It is simple to confirm this algebraically. It forms the basic building block of the cluster model. It is illustrated in QDENSITY using a density matrix example.

Another example of using QDENSITY for cluster model studies is the simplest CNOT gate, whose cluster model implementation from Ref. [13] is shown in Fig. 16.

This example is worked out in detail in *Cluster.nb*.

11 CONCLUSION

This simulation affords opportunities for many applications and extensions. The basic operations and manipulations are formulated in a modular manner and, as the illustrations and tutorial demonstrate, one can formulate and answer many important QC questions. Application to dynamical theories based on master equations, including environmental effects, are one challenge. Invoking and testing measures for entanglement and probing the role of noisy entanglement, imperfect gates, and of error correction protocols are other potential applications. Extending the study to general types of measurements and to cluster model cases is also of considerable interest. Finally, the description of real experimental situations by suitable Hamiltonians and studying the stability of QC algorithms could be an important role for future study using

49

The structure of the Cluster is essentially described in ClusterLinks, which

contains the pairs of CPHASE interacting qubits:

```
NCS = nq
ClusterLinks = {{1, 2}, {2, 3}, {2, 4}};
Nlinks = Length[ClusterLinks]
```

The 4 qubit initial state before building in the correlations is:

```
InitialState = ((ket1⊗KetX[0])⊗KetX[0])⊗ket4
ρinitial = InitialState⊗Adj[InitialState];
Chop[Tr[InitialState⊗Adj[InitialState]]]
```

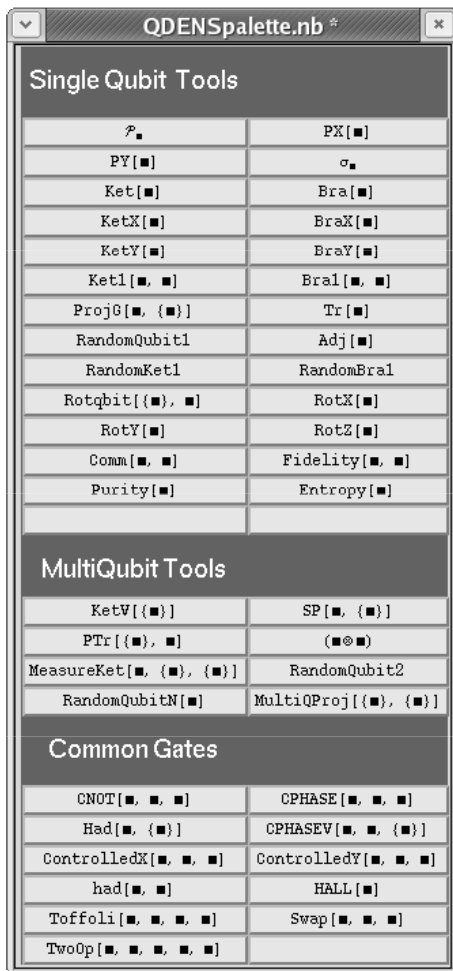Fig. 17. Example from *Cluster.nb* showing preparation of the cluster state.

QDENSITY .

QDENSpalette.nb *

Single Qubit Tools

$\mathcal{P}_\blacksquare$	PX[■]
PY[■]	$\sigma_\blacksquare$
Ket[■]	Bra[■]
KetX[■]	BraX[■]
KetY[■]	BraY[■]
Ket1[■, ■]	Bra1[■, ■]
ProjG[■, {■}]	Tr[■]
RandomQubit1	Adj[■]
RandomKet1	RandomBra1
Rotqbit[{■}, ■]	RotX[■]
RotY[■]	RotZ[■]
Comm[■, ■]	Fidelity[■, ■]
Purity[■]	Entropy[■]

MultiQubit Tools

KetV[{■}]	SP[■, {■}]
PTr[{■}, ■]	(■⊗■)
MeasureKet[■, {■}, {■}]	RandomQubit2
RandomQubitN[■]	MultiQProj[{■}, {■}]

Common Gates

CNOT[■, ■, ■]	CPHASE[■, ■, ■]
Had[■, {■}]	CPHASEV[■, ■, {■}]
ControlledX[■, ■, ■]	ControlledY[■, ■, ■]
had[■, ■]	HALL[■]
Toffoli[■, ■, ■, ■]	Swap[■, ■, ■]
TwoOp[■, ■, ■, ■, ■]	

Fig. A.1. The QDENSITY palette with the most useful commands

A   The QDENSITY palette

The package comes together with a Mathematica Palette containing most of the commands in the package and also some symbols that may be useful. A snapshot of the palette is in Fig. A.1.

Acknowledgments

References

[1] Michael A. Nielsen and Isaac I. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press (2000).

[2] http://www.wolfram.com

[3] C. P. Williams and S. H. Clearwater, "Explorations in Quantum Computing," Springer (1997); also see list on

"http://cam.qubit.org/wiki/index.php/List_of_QC_simulators" .

[4] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Phys. Rev. Lett. 70, 1895-1899 (1993).

[5] L. K. Grover, Phys. Rev. Lett. 79, 325-328 (1997).

[6] Grover's account of how he invented his algorithm is at "http://www1.bell-labs.com/user/lkgrover/papers.html" .

[7] Peter W. Shor, SIAM J. COMPUT 26 (5) 1484 (1997).

[8] J. S. Bell, Rev. Mod. Phys. 38 447 (1966).

[9] D. M. Greenberger, M. A. Horne, and A. Zeilinger, in Bell's Theorem, Quantum Theory, and Conceptions of the Universe, edited by M. Kafatos (Kluwer Academics, Dordrecht, The Netherlands, 1989), pp. 73.

[10] R. F. Werner,Phys. Rev. A 40, 4277 (1989).

[11] G. Lindblad, Commun. Math. Phys. 4 ,119 (1976).

[12] J. Preskill's Caltech remarkable lectures are available at:

http://www.theory.caltech.edu/people/preskill/ph229/ .

[13] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. 86 (2001) 5188; R. Raussendorf, D. Brownie and H.J. Briegel, Phys. Rev. A 68, 022312 (2003); Michael I. Nielsen, quant - ph/0504097

[14] John von Neumann, "Mathematical Foundations of Quantum Mechanics", Princeton University Press (1955).

[15] Edward Gerjuoy Am. J. Phys. 73 (6) 521 (2005).

[16] Circuit graphs have been drawn using I. Chuang "qasm2circ" (http://www.media.mit.edu/quanta/qasm2circ/), and Steve Flammia & Bryan Eastin "Qcircuit,"(http://info.phys.unm.edu/Qcircuit/).