- Motivation
- Current Conclusions and Schemes
- Another Important Scheme
- Future Directions
Motivation

Qubit

- Classical computers store information in bits, vs "qubits" in a Quantum computer
Motivation

Qubit
- Classical computers store information in bits, vs "qubits" in a Quantum computer

Quantum Gates
- Quantum gates are similar to logic gates in classical computing, in that they are used to manipulate a quantum system
Motivation

Qubits and Quantum Gates have Mathematical Realizations
Motivation

Qubits and Quantum Gates have Mathematical Realizations

- Qubits are vectors
Motivation

Qubits and Quantum Gates have Mathematical Realizations

- Qubits are vectors
- Quantum Gates are Unitary Matrices
Motivation

Qubits are Quantum Systems
Qubits are Quantum Systems

Letting $|0\rangle, |1\rangle$ be two measureables, the vector (qubit)

$$\begin{bmatrix} a \\ b \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

represents the superposition $a|0\rangle + b|1\rangle$
Motivation

Qubits are Quantum Systems

- Letting $|0\rangle$, $|1\rangle$ be two measureables, the vector (qubit) $\begin{bmatrix} a \\ b \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ represents the superposition $a|0\rangle + b|1\rangle$

- We concatenate 2+ qubits into multi-qubit quantum ensembles via tensor products:
Motivation

Qubits are Quantum Systems

- Letting $|0\rangle$, $|1\rangle$ be two measureables, the vector (qubit) $\begin{bmatrix} a \\ b \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ represents the superposition $a|0\rangle + b|1\rangle$

- We concatenate 2+ qubits into multi-qubit quantum ensembles via tensor products:

$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$
### Quantum Computing

- Letting $|0\rangle, |1\rangle$ be two measureables, a *qubit* $\begin{bmatrix} a \\ b \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ represents the superposition $a|0\rangle + b|1\rangle$.

- We concatenate 2+ qubits into multi-qubit quantum ensembles via tensor products:

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$$

- This 2-qubit system has 4 measureables, represented by the basis vectors of $\mathbb{C}^4$. 
Example.

Consider further

\[
\begin{bmatrix}
    a \\
    b
\end{bmatrix} \otimes
\begin{bmatrix}
    c \\
    d
\end{bmatrix} =
\begin{bmatrix}
    ac \\
    ad \\
    bc \\
    bd
\end{bmatrix}
\]
Example.
Consider further
\[
\begin{bmatrix}
  a \\
  b
\end{bmatrix} \otimes \begin{bmatrix}
  c \\
  d
\end{bmatrix} = \begin{bmatrix}
  ac \\
  ad \\
  bc \\
  bd
\end{bmatrix}
\]

- The basis vectors, corresponding to physical measureables, of the above *bipartite* or *joint* quantum state are

\[
e_1 = \begin{bmatrix}
  1 \\
  0 \\
  0 \\
  0
\end{bmatrix},
\quad
e_2 = \begin{bmatrix}
  0 \\
  1 \\
  0 \\
  0
\end{bmatrix},
\quad
e_3 = \begin{bmatrix}
  0 \\
  0 \\
  1 \\
  0
\end{bmatrix},
\quad\text{and}
\quad e_4 = \begin{bmatrix}
  0 \\
  0 \\
  0 \\
  1
\end{bmatrix}
\]
Example.

We use the physicists notation;

\[ |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \]
Example.

We use the physicists notation;

\[
|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},
|01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},
|10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},
|11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\]

So,

\[
\begin{bmatrix} ac \\
    ad \\
    bc \\
    bd \end{bmatrix} = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle
\]
Motivation

Question:
How Many Measureables does a 64-qubit multipartite system have?
Motivation

Some other operations

Let $A, B \in M_2$.

- The tensor product of $A$ and $B$ is
  \[
  A \otimes B = \begin{bmatrix}
  a_{11}B & a_{12}B \\
  a_{21}B & a_{22}B
  \end{bmatrix}.
  \]

- The direct sum of $A$ and $B$ is defined as
  \[
  A \oplus B = \begin{bmatrix}
  A & 0 \\
  0 & B
  \end{bmatrix},
  \]
  where $0 \in M_2$. 
Motivation

Quantum Gates reign things in

- An n-qubit system has $2^n$ measureable states, and a classical computer has to deal with each of these....
Motivation

Quantum gates reign things in

- An n-qubit system has $2^n$ measurable states, and a classical computer has to deal with each of these...
- A Quantum computer uses *Quantum*, or *Unitary*, Gates (Unitary matrices) to handle these n-qubit systems in a single operation.
Motivation

Definition
A matrix $U \in M_n(\mathbb{C})$ is unitary if $U \cdot U^* = U^* \cdot U = I$ where $*$ denotes the conjugate transpose.

Important Properties
- $U$ is invertible and $U^{-1} = U^*$
- The rows and columns of $U$ are orthonormal
Motivation-Example Quantum Gates in 1 qubit

Hadamard Gate

The Hadamard gate, $H$, is a commonly used gate where

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

Pauli Matrices

$$\sigma_\text{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\sigma_\text{y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$\sigma_\text{z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$
Motivation

- The set of Quantum Gates a quantum computer can generate directly determines its capability.
The set of Quantum Gates a quantum computer can generate directly determines its capability.

Obviously, we do not want to limit our systems' possible operations...
The set of Unitary Gates a quantum computer can generate directly determines its capability.

Obviously, we do not want to limit our systems’ possible operations...

We can do even better: **How can we not only allow for all operations, but have an efficient ”generating set” of simple unitaries?**
Experimentalists are working on possible physical manifestations in the 1-4 qubit cases.

2 Qubits Corresponds to 4-by-4 Unitaries
- There are two types of gates that are easy to implement
  - 1-control gates
  - Free-gates
Experimentalists are working on possible *physical* manifestations in the 1-4 qubit cases.

**2 Qubits Corresponds to 4-by-4 Unitaries**

- There are two types of gates that are easy to implement
  - 1-control gates
  - Free-gates

Experimentalists find these to be *simple* to implement.
Decomposition of Quantum Gates - 2 Qubit Case

1-Control Gates

\[(1 \mathcal{V}) = I_2 \oplus \mathcal{V} \]
\[(0 \mathcal{V}) = \mathcal{V} \oplus I_2 \]

\[(\mathcal{V}0) = \begin{bmatrix}
\nu_{11} & 0 & \nu_{12} & 0 \\
0 & 1 & 0 & 0 \\
\nu_{21} & 0 & \nu_{22} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}\]

\[(\mathcal{V}1) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \nu_{11} & 0 & \nu_{12} \\
0 & 0 & 1 & 0 \\
0 & \nu_{21} & 0 & \nu_{22}
\end{bmatrix}\]
Decomposition of Quantum Gates-2 Qubit Case

\[
(V^*) = V \otimes I_2 = \begin{bmatrix}
\nu_{11} & 0 & \nu_{12} & 0 \\
0 & \nu_{11} & 0 & \nu_{12} \\
\nu_{21} & 0 & \nu_{22} & 0 \\
0 & \nu_{21} & 0 & \nu_{22}
\end{bmatrix}
\]

\[
(*V) = I_2 \otimes V = \begin{bmatrix}
\nu_{11} & \nu_{12} & 0 & 0 \\
\nu_{21} & \nu_{22} & 0 & 0 \\
0 & 0 & \nu_{11} & \nu_{12} \\
0 & 0 & \nu_{21} & \nu_{22}
\end{bmatrix}
\]
A Decomposition Scheme-2 Qubit Case

What's the difference?

Consider a 2-qubit Vector State

\[ q = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle. \]
A Decomposition Scheme-2 Qubit Case

What’s the difference?

Consider a 2-qubit Vector State

\[ q = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle. \]

Operating on this system with a free-gate, \((V^\dagger)\), yields

\[ (I \otimes V)(q) = |0\rangle \otimes V(a_0|0\rangle + a_1|1\rangle) + |1\rangle \otimes V(a_2|0\rangle + a_3|1\rangle). \]
A Decomposition Scheme-2 Qubit Case

What's the difference?

Consider a 2-qubit Vector State

$$q = a_0 |00\rangle + a_1 |01\rangle + a_2 |10\rangle + a_3 |11\rangle.$$

Operating on this system with a free-gate, ($\ast V$), yields

$$(I \otimes V)(q) = |0\rangle \otimes V(a_0 |0\rangle + a_1 |1\rangle) + |1\rangle \otimes V(a_2 |0\rangle + a_3 |1\rangle).$$

Operating on this system with a 1-control gate, (1V), yields

$$(I \oplus V)(q) = a_0 |00\rangle + a_1 |01\rangle + |1\rangle V(a_2 |0\rangle + a_3 |1\rangle).$$
A Decomposition Scheme-2 Qubit Case

What's the difference?

**Consider a 2-qubit Vector State**

\[ q = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle. \]

Operating on this system with a free-gate, (*V), yields

\[ (I \otimes V)(q) = |0\rangle \otimes V(a_0|0\rangle + a_1|1\rangle) + |1\rangle \otimes V(a_2|0\rangle + a_3|1\rangle). \]

Operating on this system with a 1-control gate, (1V), yields

\[ (I \oplus V)(q) = a_0|00\rangle + a_1|01\rangle + |1\rangle V(a_2|0\rangle + a_3|1\rangle). \]

1-**controls**, or **controlled gates**, in general, are named so because they act solely on some of the components of a multi-partite state, and leave the rest alone (computationally expensive!)
Using control gates, one can decompose an arbitrary $n$-by-$n$ unitary matrix into a product of at most $n^2$ unitary matrices.

- P-unitary matrices are $(1V)$, $(0V)$, $(V1)$, and

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & v_{11} & v_{12} & 0 \\
0 & v_{21} & v_{22} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
$$
A Decomposition Scheme-2 Qubit case

Previous Result (Li, Roberts, Yin)

Using control gates, one can decompose an arbitrary \( n \times n \) unitary matrix into a product of at most \( \binom{n}{2} \) unitary matrices.

- P-unitary matrices are \((1V), (0V), (V1),\) and

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & v_{11} & v_{12} & 0 \\
0 & v_{21} & v_{22} & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

i.e.

\[
\binom{n}{k} = \frac{n!}{k!(n-k)!}.
\]
A Decomposition Scheme-2 Qubit case

Previous Result (Li, Roberts, Yin)

One can decompose an arbitrary $n$-by-$n$ unitary matrix into a product of at most \( \binom{n}{2} \) P-unitary matrices

- P-unitary matrices are $(1V)$, $(0V)$, $(V1)$, and

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & v_{11} & v_{12} & 0 \\
0 & v_{21} & v_{22} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

i.e.

\[
\binom{n}{k} = \frac{n!}{k!(n-k)!}.
\]

- For 4-by-4, at most 6 unitary matrices.
- For 8-by-8, at most 14 unitary matrices.
- etc.
The above decomposition scheme heavily utilized control gates. The next step was to introduce free-gates into the decomposition, and achieve a lowest possible cost.
The above decomposition scheme heavily utilized control gates. The next step was to introduce free-gates into the decomposition, and achieve a lowest possible cost.

More Important Results (Li, Pelejo)
- In the 4-by-4 case, 3 1-control gates is enough for any unitary
A Decomposition Scheme-2 Qubit case

The above decomposition scheme heavily utilized control gates. The next step was to introduce free-gates into the decomposition, and achieve a lowest possible cost.

More Important Results (Li, Pelejo)

- In the 4-by-4 case, 3 1-control gates is enough for any unitary
- We can always freely transform a 4-by-4 1-control gate into a (1V) gate
The above decomposition scheme heavily utilized control gates. The next step was to introduce free-gates into the decomposition, and achieve a lowest possible cost.

More Important Results (Li, Pelejo)

- In the 4-by-4 case, 3 1-control gates is enough for any unitary
- We can always freely transform a 4-by-4 1-control gate into a (1V) gate
- A decomposition scheme was developed and extended to all $n$, as well as a recursive formula giving the number of free and $k$-control gates that could be used to decompose an arbitrary unitary.
A Decomposition Scheme-2 Qubit case

The above decomposition scheme heavily utilized control gates. The next step was to introduce free-gates into the decomposition, and achieve a lowest possible cost.

More Important Results (Li, Pelejo)

- In the 4-by-4 case, 3 1-control gates is enough for any unitary.
- We can always freely transform a 4-by-4 1-control gate into a (1V) gate.
- A decomposition scheme was developed and extended to all \( n \), as well as a recursive formula giving the number of free and \( k \)-control gates that could be used to decompose an arbitrary unitary.
- We want(ed) to further reduce the number of controls!
Questions:

- How many gates are necessary, and, specifically, how many 1-control gates are necessary and sufficient?
- What is the most efficient scheme for decomposing general unitaries?

1-control gates are a metaphoric cost in a decomposition!
Current Scheme

How should one attack the problem?

- What can we do *for free* that simplifies the problem, or gives telling information about our candidate?
How should one attack the problem?

- What can we do *for free* that simplifies the problem, or gives telling information about our candidate? (★)
How should one attack the problem?

- What can we do *for free* that simplifies the problem, or gives telling information about our candidate? (⋆)

- Switch focus from finding ways to decompose a matrix, to finding out what must be true if the matrix can be written as a product of free gates, free gates and a single 1-control gate, etc.
If a matrix $M$ can be decomposed using only free gates, it can be written as

$$M = A \otimes B,$$

Where $A$ and $B$ are 2-by-2 unitary matrices.
If a matrix $M$ can be decomposed using only free gates, it can be written as

$$M = A \otimes B,$$

where $A$ and $B$ are 2-by-2 unitary matrices. ($\star$) This requires that each block be a scalar multiple of some unitary!
Example in 4-by-4 case

- If a matrix $M$ can be decomposed using only free gates, it can be written as
  \[ M = A \otimes B, \]
  Where $A$ and $B$ are 2-by-2 unitary matrices.  
  (*) This requires that each block be a scalar multiple of some unitary!

- If $M$ can be decomposed using free gates, and a single 1-control, then it can be written as
  \[ M = (A \otimes B)(I_2 \oplus W)(E \otimes F), \]
  Where $A, B, W, E, F$ all unitary.
Recall the Singular Value Decomposition

For any matrix $A \in M_n$, there is a unitary equivalence of $A$ yielding a diagonal matrix, with entries the singular values of $A$.
Recall the Singular Value Decomposition

*For any matrix $A \in M_n$, there is a unitary equivalence of $A$ yielding a diagonal matrix, with entries the singular values of $A$*

Example.

$$M = \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}.$$
Recall the Singular Value Decomposition

For any matrix $A \in M_n$, there is a unitary equivalence of $A$ yielding a diagonal matrix, with entries the singular values of $A$.

Example.

$$M = \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}.$$  

Its singular value decomposition yields the factorization,

$$M = U\Sigma V = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$
The unitary matrix
\[
\begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]
is known as the \textit{Not Gate}.

- It is important-a class of controlled gates utilizes its properties.

Ex., the CNOT Gate is
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}.
\]
We let \( M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \) be a general 4\( \times \)4 unitary matrix. By the SVD, there exist unitary \( U \) and \( V \) such that \( V \cdot M_{11} \cdot U = C = \text{diag}(c_1, c_2) \).

So

\[
(l_2 \otimes V) \cdot M \cdot (l_2 \otimes U) = \begin{bmatrix} C & SU \\ VS & -VCU \end{bmatrix},
\]

where \( S = \text{diag}(s_1, s_2) \).

Our Scheme revolves around the values of \( c_1 \) and \( c_2 \).
Free Decomposition (Theorem)

Given a 4 by 4 unitary matrix

\[
M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = (I_2 \otimes V)^* \begin{bmatrix} C & SU \\ VS & -VCU \end{bmatrix} (I_2 \otimes U)^*,
\]

Letting \( C = \text{diag}(c_1, c_2) \).

Then, \( M \) is a product of free gates if and only if \( c_1 = c_2 \) and \( s_1 UV^* + c_1 UV \) and \( s_1 c_1 V \) are scalar matrices.

\( \downarrow \text{i.e., for a given unitary, check three things, and you’ll know whether controlled gates are needed for decomposition!} \)
Again, take a unitary and write it as

\[ M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = (I_2 \otimes V)^* \begin{bmatrix} C & SU \\ VS & -VCU \end{bmatrix} (I_2 \otimes U)^*. \]

Then, \textit{M is a product of free gates and one 1-control gate if and only if either,}

- (i) \( c_1 = c_2 \) and \( C, S, U, \) and \( V \) are simultaneously unitarily diagonalizeable.
- (ii) \( c_1 \neq c_2 \in (0, 1) \) and \( V, U \) are both scalar matrices.
- (iii) \( C \) and \( S \) are rank 2
2 1-Control and Free Gates(?)

- We know that a unitary can be written as a product of free gates and two 1-control gates when $c_1 = c_2 \in (0, 1)$ and $U, V$ are not simultaneously diagonalizable.
- This is incomplete, $c_1 \neq c_2$ and?
The authors proved that every $U \in SU(4)$ can be written as

$$U = (A_1 \otimes A_2)(\exp(i(d_x \sigma_x \otimes \sigma_x + d_y \sigma_y \otimes \sigma_y + d_z \sigma_z \otimes \sigma_z))(B_1 \otimes B_2)$$

with $A_1, A_2, B_1, B_2 \in SU(2)$, $d_x, d_y, d_z \in \mathbb{R}$. 

[1]
We also know that any $U \in SU(4)$ is decomposable using at most three 1-control gates-[6]. **We wish to know whether the two different schemes can be used in combination.**

**i.e.**

- SVD is not computationally expensive-when is it better?
- Can this be used to find conditions where two 1-controls are sufficient?
- Insight into the general case
Future Directions

- Comparison of the two Schemes.
- Utility of Different Schemes Relative to Different Physical Manifestations.
- Find a quantitative operation on a matrix which determines which scheme is most efficient.
- Higher qubits.
Acknowledgements

- Eric Berry, Diane C. Pelejo, Dr. Chi-Kwong Li
- The NSF for their support through the EXTREEMS-QED grant
- William and Mary Mathematics
References


Chi-Kwong Li, Rebecca Roberts, Xiaoyan Yin Decomposition of unitary matrices and quantum gates arXiv:1210.7366

Chi-Kwong Li, Diane Pelejo Decomposition of quantum gates arXiv:1311.3599

M. Nakahara and T. Ohmi, Quantum Computing: From Linear Algebra to Physical Realizations