1 Introduction

A quantum circuit or quantum algorithm acting on an n-qubit system is nothing more than the action of unitary matrix on the state vector of the qubits. Up to a global phase, these operators lie in the Lie algebraic Special Unitary group $SU(2^n)$: unitary matrices of size $2^n \times 2^n$ with complex valued entries and a real determinant 1. The necessity for such operators is clear, in that any operation on n-qubits is simply a transformation from one state to another by action of a rotation through the computational space: the input vector has 2^n dimensions, and a vector with 2^n dimensions is output. Furthermore, in order to maintain the coherence of the systems, the operator must be unitary and maintain the total probability at 1.

Quantum computers do not apply any arbitrary unitary operator, and instead must select their qubit operators from a set of universal pre-defined and architecture specific implementable gates, usually a single qubit rotation operator and an entangling gate such as the CNOT or $i\sqrt{SWAP}$. Generally only unitaries acting on a pair of qubits are directly implementable, but interesting computations will require circuits and unitary operators acting on the full space of the system. A key question arises: what is the most efficient method of implementing a desired unitary operator using the minimal number of quantum circuit elements? For many algorithms, the structure of the operator allows for efficient decomposition into circuit elements, and it is often these circuits that are most promising for future applications in quantum computing. However, in the absence of an externally calculated circuit decomposition for a gate, it is still useful to know how to take any arbitrary operator and implement that gate on a quantum computer.

There are two key limiting factors for unitary decomposition, which are essential for understanding the useful application and scope of these methods. First, for an n-qubit unitary matrix, the CNOT gate depth scales as $\mathcal{O}(4^n)$. The scaling here is proven to be minimal, so while certain algorithms improve the coefficient of the 4^n term, in the worst case and given no prior knowledge of a unitary, the computation time scales exponentially with the number of qubits. Essentially this means that not all circuits will provide a speed-up over a classical system, because it can take exponential time to run a quantum circuit. Second, the current best algorithms for decomposing a unitary matrix are necessarily more computationally expensive than simply simulating the quantum circuit. Specifically, methods such as the cosine sine decomposition (CSD) require calculating the eigenvalues of the unitary operator, which is often what the quantum circuit would be used to find in the first place. However, given a proper understanding of the constraints of arbitrary unitary matrix decomposition, there are plenty of useful applications for the algorithms, especially in near term applications. For instance, the most efficient construction of the circuit for an arbitrary gate $U \in SU(2)$ or $U \in SU(4)$ are both well understood and easily implementable, and as such are valuable as elementary components in larger circuits. Although it is impractical to use arbitrary decomposition on systems larger than a few dozen qubits, it can be useful as a tool for implementing new algorithms on limited scale test circuits before an efficient method of decomposition has been discovered. Arbitrary decomposition assumes very little about the structure of the operator, so it is generally possible to find a much more efficient decomposition by hand, albeit this is likely not a simple task. Additionally, arbitrary unitary decomposition provides a maximal circuit depth for any algorithm operating on an n-qubit system. For example, any algorithm on two qubits which would naively require 10 CNOT gates and 30 single qubit rotations can always be optimized to at most require 3 CNOT and 18 single qubit rotations [4, 8].

In this paper, we present three key results in unitary decomposition. First, we present a primitive algorithm which proves the universality of single qubit operations and CNOT gates. Next, we present a constructive algorithm to iteratively construct a near best-case and easily implementable decomposition. Finally, we present key results for the Cartan decomposition construction of a universal circuit, which is the foundation of the currently best known constructive decomposition algorithm [2].

2 Universal Gates

To construct our universal basis of gates, we will confine ourselves to the following physically implementable gate operators: $R_Z(\theta)$, an arbitrary rotation around the Z axis by an angle θ, $R_y(\alpha)$, an arbitrary rotation around the

Y axis by an angle α, and the Controlled-Not gate, which have matrix representations

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \qquad R_y(\alpha) = \begin{bmatrix} \cos(\frac{\alpha}{2}) & -\sin(\frac{\alpha}{2}) \\ -\sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) \end{bmatrix} \qquad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1}$$

These operators are universal, as we will demonstrate, but they are by no means the only universal elements. In particular, either R_z or R_y can be replaced with R_x without affecting any of the following results. Additionally, these operations represent operations on a single qubit or pair of qubits but are only physically implementable on physical, not logical, qubits. Universal gates for logical qubits are not addressed in this paper.

3 Givens Rotations

3.1 Decomposition into Givens Rotations

The QR decomposition theory states that any complex operator A can be decomposed into an orthogonal operator Q and an upper triangular matrix R as A = QR. Essential to our application is that the operator Q can be further decomposed as a sequence of 2D Givens rotations, which apply a rotation θ in the i^{th} and j^{th} plane as

$$V(i,j,\theta) = \begin{bmatrix} \ddots & & & & \iddots \\ & c = cos(\theta)_{ii} & 0 & -s = -sin(\theta)_{ji} & \\ & 0 & 1 & 0 & \\ & s = sin(\theta)_{ij} & 0 & c = cos(\theta)_{jj} & \\ \iddots & & & & \ddots \end{bmatrix} \tag{2}$$

A d-dimensional square matrix Q can be decomposed into at most $\frac{d(d-1)}{2}$ of these two-level Givens rotations, and for a unitary matrix $A = U$, we can additionally assert that R must be a unitary upper triangular matrix, which is diagonal. So, for a unitary operator U, it is possible to construct a sequence of Givens rotations V_n such that $U = V_1V_2V_3V_4...V_{d(d-1)/2}R$ [1, 6, 5]. For an n-qubit system, U has dimension 2^n and a sequence of $\frac{1}{2}(4^n - 2^n) \approx \mathcal{O}(4^n)$ Givens rotations are required to construct it.

3.2 Constructing Givens Rotations using Elementary Circuit Elements

We must now concern ourselves with the construction of the Givens rotations using our chosen universal basis. Following the formulation used by Nielsen and Chuang [6], we first note that for a rotation $V(i,j,\theta)$, V only acts non-trivially on the two binary basis vectors with non-zero elements at the i^{th} and j^{th} positions, which we will call $|i\rangle$ and $|j\rangle$. A Grey code is a sequence of binary numbers, starting at the binary sequence for i and ending at j, which changes only by one place each step. For the 4×4 matrix

$$U = \begin{bmatrix} c & 0 & 0 & -s \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s & 0 & 0 & c \end{bmatrix} \tag{3}$$

We have $|i\rangle = |11\rangle$ and $|j\rangle = |00\rangle$. So, as the simplest example, the Grey code is the sequence

A	B
0	0
0	1
1	1

Table 1: Grey code sequence for decomposing U in Equation 3

We can implement the circuit by performing a multi-controlled operation controlled by the static elements of each step of the Grey code and targeting the changing element with an X operation. This is repeated for all but the final step, during which we perform the operation of the submatrix $\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = R_y(2\theta)$ controlled by the state of the remaining circuit elements. Finally, we undo the operations to return the system to $|i\rangle$.

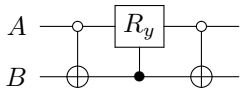


Figure 1: Circuit Implementation of the Givens rotation in Equation 3

Other results show that multi-controlled gates can be constructed using $\mathcal{O}(n)$ CNOTs and single qubit rotations [6], the grey code will have at most n-1 terms, and R, a diagonal matrix, can be efficiently implemented [5]. Thus, the total CNOT count to implement an arbitrary unitary is of the order $\mathcal{O}(n^2 4^n)$. This is by no means an optimal solution, but it is sufficient to demonstrate the universality of the single and two qubit gates.

3.3 Implementation Using Cosine Sine Decomposition

Using two decompositions, the cosine sine decomposition and quantum Shannon decomposition, we now demonstrate how to construct a decompsoition with $\mathcal{O}(4^n)$ CNOT efficiency. It is used to factorize an arbitrary unitary matrix on n qubits in a way that is easily calculable using existing optimal methods for performing singular value decomposition and eigenvalue decomposition. An in-depth exploration of the method can be found in Krol [4], and we will summarize the key results here.

First, we consider the quantum Shannon decomposition [7], which allows for a recursive calculation of the decomposition for an n-qubit operator into a sequence of four (n-1)-qubit operators and three multi-controlled single qubit rotations on the nth qubit. From there, the (n-1)-qubit operators can be again decomposed until they are trivial to implement using the single qubit operator decomposition that we demonstrate in Section 5. Figure 2 shows the basic circuit decomposition applied to a single step.

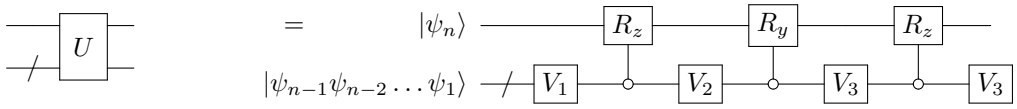
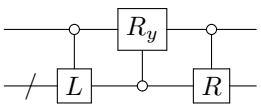


Figure 2: Quantum Shannon Decomposition [7]

It remains to calculate the values for the V_n and controlled rotations, for which we introduce the cosine sine decomposition on the matrix n-qubit operator U

$$U = \underbrace{\begin{bmatrix} R_0 & 0 \\ 0 & R_1 \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} C & -S \\ S & C \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} L_0 & 0 \\ 0 & L_1 \end{bmatrix}}_{L} \tag{4}$$

where C and S are real diagonal matrices satisfying $C^2 + S^2 = I$ and L_0, L_1, R_0, and R_1 are square unitary matrices of dimension 2^{n-1}. The formulation for R (and L) is such that it is a uniformly controlled multi-qubit operator controlled by the n^{th} qubit and performing either R_0 or R_1. Additionally, D is a multi-controlled operator which acts as a single Y rotation θ on the n^{th} qubit. So, we can write the circuit diagram for U as



It's not immediately clear that this circuit takes the same form as quantum Shannon, but we can do additional processing on L and R to show these two formulations are equivalent.

We now apply singular value decomposition (SVD) to evaluate the terms in the matrices. The SVD of a matrix U is $U = A\Sigma B^\dagger$, where A and B are unitary matrices and Σ is a diagonal matrix. SVD is a generally useful algorithm, and as such is well implemented and researched for a variety of methods and a realistic method to use and implement for a real quantum circuit. The value of SVD comes from manipulation of the terms in the cosine sine decomposition

$$U = \begin{bmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{bmatrix} = \begin{bmatrix} R_0 C L_0 & -R_0 S L_1 \\ R_1 S L_0 & R_1 C L_1 \end{bmatrix} \tag{5}$$

Setting these terms equal element wise, we produce four equations, one for each $U_{(0,1)(0,1)}$ term. The terms then bear a striking resemblance to the form of SVD, and we are able to now calculate these values from the results of A, B, and Σ applied to one of the four sub-matrices.

Finally, we note that because L and R are uniformly controlled gates, we can decompose the gate as

(6)

so that it is physically implementable with W and V being unitary matrices 2^{n-1} dimensional. This is done using the eigenvalue decomposition

$$L = \begin{bmatrix} L_0 & 0 \\ 0 & L_1 \end{bmatrix} = \begin{bmatrix} V & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & D^\dagger \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \tag{7}$$

Along with the cosine sine decomposition terms, these decompositions combine to give the quantum Shannon decomposition, which can then be repeated on the V_n operators of dimension 2^{n-1}. This is the key result of this paper, as we now have a calculable and implementable arbitrary unitary decomposition into gates which are easily implementable to a quantum computer. Although it is not the best know algorithm to decompose the unitary, it uses only $\mathcal{O}(4^n)$ CNOT operations, as opposed to the $\mathcal{O}(n^2 4^n)$ CNOTs required by the basic decomposition into Givens Rotations. Theoretically, the minimum gate decomposition approaches a CNOT count of $(\frac{1}{4})(4^n - 3n - 1)$, and the decomposition algorithm shown here is shown to use $\frac{3}{4}4^n - \frac{3}{2}2^n$ CNOT gates [4].

4 Cartan-Based Decomposition

It is also possible to decompose these circuits by exploiting the properties of the Lie group $SU(2^n)$. Specially, we make use of a form of the Cartan decomposition, for which we will briefly summarize the fundamental structures. Unfortunately there is not space here to elaborate on the fundamentals of Lie groups, but we will address results for the decomposition of a unitary operator $G \in SU(2)$ which are the basis for a Cartan based quantum Shannon decomposition.

4.1 Background and Notation

$SU(2^n)$, the group of $2^n \times 2^n$ complex unitary matrices with determinant 1 has an associated Lie algebra $\mathfrak{su}(2^n)$, which means any element $G \in SU(2^n)$ can be generated by some element $g \in \mathfrak{su}(2^n)$ as $e^{i \cdot g}$. As a basis for $\mathfrak{su}(2^n)$, we use the space spanned by the permutations of the tensor products of the Pauli matrices and the identity matrix, called Pauli strings, as we show in Table 2. The number of elements in the basis scales as 4^n, as there are three Pauli matrices and one identity matrix available for each qubit in the tensor product. As a brief note on notation, we use

$$X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad I = \sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{8}$$

and truncate Pauli strings such as $X \otimes Y \otimes Z \otimes I$ to $XYZI$, writing the product of two Pauli matrices as $X \cdot Y$. The Lie Bracket in $\mathfrak{su}(2^n)$ is the commutator relationship $[A, B] = A \cdot B - B \cdot A$, and we make use of the commutator identities $[a, b] = -[b, a]$ and

$$[X, Y] = 2iZ \qquad [Y, Z] = 2iX \qquad [Z, X] = 2iY \qquad [X, X] = 0 \qquad [I, X] = 0 \tag{9}$$

Finally, the commutator acts component-wise on the tensor product as $[I \otimes Y, Z \otimes X] = I \cdot Z \otimes Y \cdot X - Z \cdot I \otimes X \cdot Y = Z \otimes -2iZ = -2iZZ$.

$\mathfrak{su}(2)$	X	Y	Z	
$\mathfrak{su}(4)$		IX	IY	IZ
	XI	XX	XY	XZ
	YI	YX	YY	YZ
	ZI	ZX	ZY	ZZ
$\mathfrak{su}(2^n)$	$\mathfrak{su}(2^{2-1}) \otimes I$	$\mathfrak{su}(2^{2-1}) \otimes X$	$\mathfrak{su}(2^{2-1}) \otimes Y$	$\mathfrak{su}(2^{2-1}) \otimes Z$

Table 2: Pauli String Basis for $\mathfrak{su}(2^n)$

4.2 Cartan Decomposition

A Cartan decomposition of a Lie algebra $\mathfrak{g}$ is an orthogonal split of $\mathfrak{g}$ such that $\mathfrak{g} = \mathfrak{k}^1 \oplus \mathfrak{m}$ satisfying

$$[\mathfrak{k}, \mathfrak{k}] \subset \mathfrak{k}, \qquad [\mathfrak{m}, \mathfrak{m}] \subset \mathfrak{k}, \qquad [\mathfrak{k}, \mathfrak{m}] \subset \mathfrak{m} \tag{10}$$

A element $\mathfrak{h}$ is considered a Cartan subalgebra of $\mathfrak{g}$ if it is a maximal Abelian group $\mathfrak{h} \subset \mathfrak{m}$: a largest possible set of elements in $\mathfrak{m}$ for which $[a, b] = \mathbf{0}$ with $a, b \in \mathfrak{h}$. A well-known property of the Cartan subalgebras is [3]: **Property 1**: There is some group element $K \in \mathbf{K}$, the space spanned by $e^{i\mathfrak{k}}$, and some algebra element $h \in \mathfrak{h}$ for which any element $m \in \mathfrak{m}$ can be decomposed into $KhK^\dagger = m$.

A Cartan decomposition of the group $\mathbf{G}$ is given as

$$\mathbf{G} = \mathbf{KHK} \tag{11}$$

for $\mathbf{H} = e^{i\mathfrak{h}}$ and $\mathbf{K} = e^{i\mathfrak{k}}$ [3]. Additionally, a single element $G \in \mathbf{G}$ has a Cartan decomposition of the form

$$G = K_0 M = K_0 K_1 H K_1^\dagger = e^{k_0} e^{k_1} e^h e^{-k_1} = e^{k_0} e^m \tag{12}$$

Note, the element M can be decomposed as $M = K_0 K_1 H K_1^\dagger$ due to Property 1, which specifies that an element m, which generates M as e^{im}, can be written as $KhK^\dagger$, and the additional property that $\exp(KhK^\dagger) = KHK^\dagger$.

5 Universal 2-qubit Circuit

The basic example of decomposing a system using Cartan decomposition is for a single qubit operator $G \in SU(2)$ generated by $g \in \mathfrak{su}(2)$. The operator, as we can see on the Bloch sphere is equivalent to an arbitrary rotation over the surface of a unit sphere, is parameterized by coefficients for each of the three terms in the Pauli basis of $\mathfrak{su}(2) = \text{span}\{X, Y, Z\}$. That is, any operator in $G \in SU(2)$ can be written as $G = e^{aiX+biY+ciZ}$. Using quantum circuits, it is simple to represent any matrix of the form exp(i*(a single Pauli string)) using 2n CNOTs [6], but the exponential of the sum of non-commuting Pauli matrices is much less simple to evaluate on a circuit. Nominally, we would separate the terms in the vector g, but with non-commuting terms in the exponential, we are not able to use the standard identity for commuting a and b $e^{a+b} = e^a e^b$. Instead, we need to find a different method to split G into the product of exponential-of-single-Pauli terms.

For SU(2), we can classify the Pauli basis into a Cartan decomposition by hand, noting that the commutation of any two Pauli matrices is the third, and the commutation of that third matrix with any of the first two is the other matrix. Using equation 10, we can see that arbitrarily choosing X to be in $\mathfrak{m}$ and choosing Z in $\mathfrak{k}$, we have $[X, Z] = -2iY \in \mathfrak{m}$ according to the relationships of equation 10. We can verify that $[Y, X] \in \mathfrak{k}$, so $\mathfrak{k} = \text{span}\{Z\}$ and $\mathfrak{m} = \text{span}\{Y, X\}$. X and Y do not commute, so we can select $\mathfrak{h} = X$ or Y: in this case we choose $\mathfrak{h} = \text{span}\{Y\}$ to align with our earlier choice of R_z and R_y rotation gates as universal. From equation 12, there exist three elements, k_0, h, and k_1, which together give $G = e^{ik_0} e^{ik_1} e^{ih} e^{-ik_1} = e^{ik_2} e^{ih} e^{-ik_1} = e^{i\alpha Z} e^{i\beta Y} e^{-i\gamma Z} = R_z(2\alpha) R_y(2\beta) R_z(2\gamma)$. So, we have shown that any single qubit operator, up to a global phase, can be decomposed into a sequence of only three single qubit rotations. An algorithm to perform this decomposition can be found in Drury and Love, along with, at the time of writing and to the best of my knowledge, the most efficient known constructive algorithm for unitary decomposition using the Shannon Decomposition [2].

[1] Fraktur font for the letter k

References

[1] George Cybenko. "Reducing quantum computations to elementary unitary operations". In: *Computing in Science and Engineering* 3.2 (2001), pp. 27–32. ISSN: 15219615. DOI: 10.1109/5992.908999.

[2] Byron Drury and Peter Love. "Constructive quantum Shannon decomposition from Cartan involutions". In: *Journal of Physics A: Mathematical and Theoretical* 41.39 (2008), p. 13. ISSN: 17518113. DOI: 10.1088/1751-8113/41/39/395305.

[3] Navin Khaneja and Steffen Glaser. *Cartan Decomposition of SU(2 n), Constructive Controllability of Spin systems and Universal Quantum Computing*. Tech. rep. 2000. URL: http://arxiv.org/abs/quant-ph/0010100.

[4] Am Krol September. "Unitary Decomposition Implemented in the OpenQL programming language for quantum computation". In: (). URL: http://repository.tudelft.nl/..

[5] M. Mottonen and J. J. Vartiainen. *Decompositions of general quantum gates*. Tech. rep. 2005. URL: http://arxiv.org/abs/quant-ph/0504100.

[6] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. 10th Anniv. Cambridge University Press, 2010. ISBN: 9781107002173. URL: www.cambridge.org.

[7] Vivek V Shende, Stephen S Bullock, and Igor L Markov. *Synthesis of Quantum Logic Circuits*. Tech. rep. 2006.

[8] Vivek V Shende, Igor L Markov, and Stephen S Bullock. *Minimal Universal Two-Qubit CNOT-based Circuits*. Tech. rep. 2004.