Optimal ancilla-free Pauli+$V$ approximation of $z$-rotations

Neil J. Ross
Department of Mathematics and Statistics
Dalhousie University

Abstract

We describe a new efficient algorithm to approximate $z$-rotations by ancilla-free Pauli+$V$ circuits, up to a given precision $\varepsilon$. Our algorithm is optimal in the presence of an oracle for integer factoring: it outputs the shortest Pauli+$V$ circuit solving the given problem instance. In the absence of such an oracle, our algorithm is still near-optimal, producing circuits of $V$-count $m + O(\log(\log(1/\varepsilon)))$, where $m$ is the $V$-count of the third-to-optimal solution. Our method is based on previous work by Ross and Selinger on the optimal ancilla-free approximation of $z$-rotations using Clifford+$T$ gates and on previous work by Bocharov, Gurevich, and Svore on the asymptotically optimal ancilla-free approximation of $z$-rotations using Pauli+$V$ gates.

1 Introduction

1.1 The synthesis problems

The unitary group of order 2, denoted $U(2)$, is the group of $2 \times 2$ complex unitary matrices. We also refer to the elements of this group as gates. The special unitary group of order 2, denoted by $SU(2)$, is the subset of $U(2)$ consisting of unitary matrices of determinant 1. We will be concerned with the notion of distance between these matrices that arises from the operator norm, that is:

$$\|U - U'\| = \sup \{|Uv - U'v| : |v| = 1\}.$$

We refer to subsets of $U(2)$ as gate bases and to a finite word $W$ over a gate base $B$ as a circuit over $B$. By a slight abuse of notation, we write $W$ to denote both a circuit over $B$ and the unitary obtained by multiplying the basis elements composing $W$.

We are interested in decomposing, or synthesizing, unitary matrices into circuits over a given gate base. For a gate base $B$ and unitary matrix $U$, the decomposition of $U$ over $B$ can be done exactly, if there exists a circuit $W$ over $B$ such that $W = U$, or approximately up to some $\varepsilon > 0$, if there exists a circuit $W$ over $B$ such that $\|U - W\| \leq \varepsilon$. We thus get the following two problems.

- **Exact synthesis problem for $B$**: given a unitary $U$, determine whether there exists a circuit $W$ over $B$ such that $W = U$ and, in case there is, find such a circuit.

- **Approximate synthesis problem for $B$**: given a unitary $U$ and a precision $\varepsilon \geq 0$, determine whether there exists a circuit $W$ over $B$ such that $\|W - U\| \leq \varepsilon$ and, in case there is, find such a circuit.

In what follows, we focus on finite gate bases. If $B$ is such a gate base, then the set of circuits over $B$ is countable. Since $U(2)$ is uncountable, this implies that the exact synthesis problem for $B$ will sometimes be solved negatively: there are unitary matrices that cannot be exactly synthesized over $B$. However, if $B$ is universal for quantum computing, then by definition the set of circuits over $B$ is dense in $U(2)$. In this case, the approximate synthesis problem for $B$ can always be solved positively.

Because the state of a qubit is defined up to scaling by a unit scalar, the synthesis of a unitary $U$ is sometimes done up to a phase. This means that instead of finding a circuit $W$ such that $\|U - W\| \leq \varepsilon$, one looks for a circuit $W$ and a unit scalar $\lambda$ such that $\|U - \lambda W\| \leq \varepsilon$. This defines a third synthesis problem.

- **Approximate synthesis problem for $B$ up to a phase**: given a unitary $U$ and a precision $\varepsilon \geq 0$, determine whether there exists a circuit $W$ over $B$ and a unit scalar $\lambda$ such that $\|U - \lambda W\| \leq \varepsilon$ and, in case there is, find such a circuit.
Because a global phase has no observable effect in quantum mechanics, it is often sufficient to define a decomposition method for special unitary matrices. Indeed, suppose that $B$ is a gate base such that the set of circuits over $B$ is dense in $SU(2)$. If we have an algorithm to approximately synthesize elements of $SU(2)$ into circuits over $B$, then we can synthesize arbitrary unitary matrices over $B$ up to a phase, since the determinant of a unitary matrix always has norm 1.

A decomposition method solving any of the above three problems is evaluated with respect to its time complexity (what is its run-time?) and to its circuit complexity (how many gates are contained in the produced circuit?).

1.2 Synthesis of $z$-rotations in the $V$-basis

We are interested in the following $V$ gates:

$$V_X = \frac{1}{\sqrt{5}}(I + 2iX) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2i \\ 2i & 1 \end{pmatrix}, \quad V_Y = \frac{1}{\sqrt{5}}(I + 2iY) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix},$$

and their adjoints:

$$V_{X}^\dagger = \frac{1}{\sqrt{5}}(I - 2iX) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2i \\ -2i & 1 \end{pmatrix}, \quad V_{Y}^\dagger = \frac{1}{\sqrt{5}}(I - 2iY) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix},$$

$$V_{Z}^\dagger = \frac{1}{\sqrt{5}}(I - 2iZ) = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 0 \\ 0 & 1 + 2i \end{pmatrix}.$$

It was shown in [6] and [7] that the group generated by the $V$ gates is dense in $SU(2)$. It was later shown in [5] that for any operator $U \in SU(2)$ and any precision $\varepsilon$, there exists an approximation for $U$ over $V = \{V_X, V_Y, V_Z, V_X^\dagger, V_Y^\dagger, V_Z^\dagger\}$ that requires only $O(\log(1/\varepsilon))$ gates. However, no approximate synthesis algorithm was provided. In [1], Bocharov, Gurevich, and Svore defined a probabilistic algorithm for the approximate synthesis of unitaries over the Pauli+$V$ gate set, consisting of the $V$ gates together with the usual Pauli gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \text{and,} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In the context of the Pauli+$V$ gate set, the complexity of a circuit is measured by counting the number of $V$ gates appearing in it, its $V$-count. This is due to the fact that the Pauli operators can always be moved to the end of a circuit using equations such as $XV_X = V_X X$, $YV_Y = V_Y Y$, $ZV_Z = V_Z Z$ and so on.

The algorithm of [1] is efficient in the sense that it runs in probabilistic polynomial time. Moreover, it yields circuits of $V$-count bounded above by $12 \log_5(2/\varepsilon)$ for arbitrary unitaries.

The method of [1] was adapted from the one developed in [10] for the Clifford+$T$ gate set. It relies on the definition of an algorithm for the Pauli+$V$ decomposition of $z$-rotations, i.e., matrices of the form

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

For such matrices, the algorithm of [1] achieves circuits of $V$-count bounded above by $4 \log_5(2/\varepsilon)$. Such an algorithm can then be used for the synthesis of an arbitrary element $U$ of $SU(2)$ by first writing $U$ as a product of three $z$-rotations using Euler angles

$$U = R_z(\theta_1)X R_z(\theta_2)XR_z(\theta_3)$$

and then applying the algorithm to each of the $R_z(\theta_i)$.

1.3 Results

In this paper, we define an efficient and optimal algorithm for the approximate synthesis of $z$-rotations over the Pauli+$V$ gate set. Our algorithm is defined by adapting techniques developed in [9] for the Clifford+$T$ gate set. We stress that the algorithm is literally optimal, i.e., for any given pair $(\theta, \varepsilon)$ of an angle and a precision, the algorithm finds the shortest possible ancilla-free Pauli+$V$ circuit $W$ such that $\|W - R_z(\theta)\| \leq \varepsilon$. As in [9], the optimality of the algorithm depends on the presence of a factoring oracle. Because of Shor’s algorithm [11], a quantum computer can serve as such an oracle. For this reason, the algorithm is actually an efficient and optimal quantum synthesis algorithm. However, the classical algorithm obtained in the absence of a factoring oracle is still near-optimal; in this case the algorithm produces circuits of $V$-count $m + O(\log(\log(1/\varepsilon)))$, where $m$ is the $V$-count of the third-to-optimal solution.
2 Preliminaries

We write \( \mathbb{N} \) for the semiring of non-negative integers, \( \mathbb{Z} \) for the ring of integers and \( \mathbb{C} \) for the field of complex numbers. The conjugate of a complex number is given by \((a + ib)^\dagger = a - ib\). Recall that the Gaussian integers \( \mathbb{Z}[i] \) are the complex numbers whose real and imaginary parts are both integral, i.e., the complex numbers \( a + ib \) with \( a, b \in \mathbb{Z} \). The units of \( \mathbb{Z}[i] \) are \( \pm 1, \pm i \).

3 Exact synthesis in the Pauli+V gate set

In this section, we describe an algorithm to solve the problem of exact synthesis in the Pauli+V gate set. This material is adapted from [1], where a very similar algorithm was described using the theory of quaternions. We also use some techniques developed in [3] for exact synthesis in the Clifford+T gate set. The results of this section are not new and are only included for completeness.

Problem 3.1. Given \( U \in U(2) \), determine whether there exists a circuit \( W \) over the Pauli+V gate set such that \( U = W \) and, in case there is, find such a circuit with minimal V-count.

To solve Problem 3.1 we consider unitary matrices of the form

\[
U = \frac{1}{\sqrt{5^k}} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},
\]

where \( k \in \mathbb{N} \) and \( \alpha, \beta, \gamma, \delta \in \mathbb{Z}[i] \). Each integer \( k \) such that \( U \) can be written as in (1) is called a denominator exponent of \( U \). The least such \( k \) is the least denominator exponent of \( U \). The notions of denominator exponent and of least denominator exponent extend naturally to vectors and scalars of the form

\[
\frac{1}{\sqrt{5^k}} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix} \quad \text{and} \quad \frac{1}{\sqrt{5^k}} \alpha,
\]

where \( k \in \mathbb{N} \) and \( \alpha, \gamma \in \mathbb{Z}[i] \).

Lemma 3.2. Let \( u \) be a 2-dimensional unit vector of the form (2) with least denominator exponent \( k \). Then there exists a Pauli+V circuit \( W \) of V-count \( k \) such that \( Wu = e_1 \), the first standard basis vector.

Proof. Write \( u \) as in (2) with \( \alpha = a + ib \) and \( \gamma = c + id \):

\[
u = \frac{1}{\sqrt{5^k}} \begin{pmatrix} a + ib \\ c + id \end{pmatrix}.
\]

Note that, since \( u \) has unit norm, we have \( a^2 + b^2 + c^2 + d^2 = 5^k \). We now prove the lemma by induction on \( k \).

- \( k = 0 \). In this case \( a^2 + b^2 + c^2 + d^2 = 1 \). It follows that exactly one of \( a, b, c, d \) is \( \pm 1 \) while all the others are 0. It is then easy to show that \( u \) can be reduced to \( e_1 \) by acting on it using a Pauli operator.

- \( k > 0 \). In this case \( a^2 + b^2 + c^2 + d^2 \equiv 0 \pmod{5} \). We will show that there exists a Pauli+V operator \( U \) of V-count 1 such that the least denominator exponent of \( Uu \) is \( k - 1 \). It then follows by the induction hypothesis that there exists \( U' \) of V-count \( k - 1 \) such that \( U'Uu = e_1 \), which then completes the proof.

Consider the residues, modulo 5, of \( a, b, c, \) and \( d \). Since 0, 1, and 4 are the only squares modulo 5, then, up to a reordering of the tuple \((a, b, c, d)\), we must have:

\[
(a, b, c, d) \equiv \begin{cases} 
(0, 0, 0, 0) \\
(\pm 2, \pm 1, 0, 0) \\
(\pm 2, \pm 2, \pm 1, \pm 1).
\end{cases}
\]

However, by minimality of \( k \), we know that \( a \equiv b \equiv c \equiv d \equiv 0 \) is impossible, so the other two cases are the only possible ones. We treat them in turn.

First, assume that one of \( a, b, c, d \) is congruent to \( \pm 2 \), one is congruent to \( \pm 1 \) and the remaining two are congruent to 0. By acting on \( u \) with a Pauli operator, we can moreover assume without loss of generality that \( a \equiv 2 \). Now if \( b \equiv 1 \), consider \( V_2u \):

\[
V_2u = \frac{1}{\sqrt{5^{k+1}}} \begin{pmatrix} (a - 2b) + i(2a + b) \\
(c + 2d) + i(d - 2c) \end{pmatrix}.
\]
Let \( \text{Proposition 3.4.} \)

Let \( U \) and \( V \) be such that the first column of \( W \) is of the form \((a, b, c, d)\). Now assume that two of \( a, b, c, d \) are congruent to \( \pm 2 \) while the remaining two are congruent to \( \pm 1 \). We can use Pauli operators to guarantee that \( a \equiv 2 \) and \( c \geq 0 \). As above, we list the desired operators in a table for conciseness. It can easily be checked that in each case the given operator is such that the least denominator exponent of \( Uu \) is \( k - 1 \).

| \((a, b, c, d)\) | \( U \)          |
|------------------|----------------|
| \((2, 1, 0, 0)\) | \( V_Z \)      |
| \((2, 0, 1, 0)\) | \( V_Y \)      |
| \((2, 0, 0, 0)\) | \( V_X \)      |
| \((2, -1, 0, 0)\) | \( V_Z \)      |
| \((2, 0, -1, 0)\) | \( V_Y \)      |
| \((2, 0, 0, -1)\) | \( V_X \)      |

Now assume that two of \( a, b, c, d \) are congruent to \( \pm 2 \) while the remaining two are congruent to \( \pm 1 \). We can use Pauli operators to guarantee that \( a \equiv 2 \) and \( c \geq 0 \). As above, we list the desired operators in a table for conciseness. It can easily be checked that in each case the given operator is such that the least denominator exponent of \( Uu \) is \( k - 1 \).

| \((a, b, c, d)\) | \( U \)          |
|------------------|----------------|
| \((2, 2, 1, 1)\) | \( V_Y \)      |
| \((2, 1, 2, 1)\) | \( V_X \)      |
| \((2, 1, 1, 2)\) | \( V_Z \)      |
| \((2, 1, 2, -1)\) | \( V_Z \)      |
| \((2, -1, 2, 1)\) | \( V_Z \)      |
| \((2, 2, 1, -1)\) | \( V_Y \)      |
| \((2, -2, 1, 1)\) | \( V_X \)      |
| \((2, 1, 1, -2)\) | \( V_Y \)      |
| \((2, -1, 1, 2)\) | \( V_Y \)      |
| \((2, -1, 1, -2)\) | \( V_Z \)      |
| \((2, -2, 1, -1)\) | \( V_X \)      |
| \((2, -1, 2, -1)\) | \( V_Y \)      |

**Corollary 3.3.** Let \( U \) be a unitary matrix of determinant \( \pm 1 \) of least denominator exponent \( k \). Then there exists a Pauli+V circuit \( W \) of V-count \( k \) such that \( W = U \).

**Proof.** It suffices to show that there exists a Pauli+V circuit \( W' \) of V-count \( k \) such that \( W'U = I \) because we can then let \( W = W'^† \). To obtain such a circuit, apply Lemma \( 3.2 \) to the first column \( u_1 \) of \( U \). This yields a circuit \( W' \) such that the first column of \( W'U \) is \( e_1 \). Since \( W'U \) is unitary, it follows that its second column \( u_2' \) is a unit vector orthogonal to \( e_1 \). Therefore \( u_2' = \lambda e_2 \) where \( \lambda \) is a unit of the Gaussian integers. Since the determinant of \( W'U \) is \( \pm 1 \), it follows that \( \lambda = \pm 1 \). Thus either \( W'U = I \) or \( ZW'U = I \) and this completes the proof.

We can now solve Problem \( 3.4 \).

**Proposition 3.4.** Let \( U \in U(2) \). Then \( U \) is exactly representable by a Pauli+V operator if and only if \( \det(U) = \pm 1 \) and \( U \) is of the form

\[
U = \frac{1}{\sqrt{5^k}} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},
\]

with \( k \in \mathbb{N} \) and \( \alpha, \beta, \gamma, \delta \in \mathbb{Z}[i] \). Moreover, there exists an efficient algorithm for computing a Pauli+V circuit \( W \) for \( U \) whose V-count is equal to the least denominator exponent of \( U \), which is minimal.
Proof. For the first claim, the left-to-right implication holds because the generators of Pauli+$V$ are of the desired form, which is preserved by multiplication. The converse implication follows from Corollary 3.3. Now suppose that the least denominator exponent of $U$ is $k$. Then $W$ can be efficiently computed because the algorithm described in the proofs of Lemma 3.2 and Corollary 3.3 requires $O(k)$ arithmetic operations. Moreover, Corollary 3.3 implies that the $V$-count of $W$ is $k$. Finally, $k$ is minimal, because clearly any Pauli+$V$ circuit of $V$-count up to $k-1$ has least denominator exponent at most $k-1$. \hfill \qed

4 Approximate synthesis of $z$-rotations over the Pauli+$V$ gate set

In this section, we describe an algorithm to solve the problem of approximate synthesis of $z$-rotations over the Pauli+$V$ gate set.

Problem 4.1. Given an angle $\theta$ and a precision $\varepsilon > 0$, find a Pauli+$V$ circuit $U$ such that $||R_z(\theta) - U|| \leq \varepsilon$ and the $V$-count of $U$ is as small as possible.

Our algorithm is adapted from the one developed in [9] for the Clifford+$T$ gate set. As in [9], we reduce Problem 4.1 to a pair of independent problems. From Proposition 3.4 we know that a unitary matrix $U$ can be efficiently decomposed as a Pauli+$V$ circuit if and only if

\[
U = \frac{1}{\sqrt{5}} \begin{pmatrix}
\alpha & -\beta^* \\
\beta & \alpha^*
\end{pmatrix}, \quad \text{with } k \in \mathbb{N}, \alpha, \beta, \gamma, \delta \in \mathbb{Z}[i], \text{ and } \det(U) = \pm 1. \tag{3}
\]

To solve Problem 4.1 we therefore need to find $k \in \mathbb{N}$ and $\alpha, \beta, \gamma, \delta \in \mathbb{Z}[i]$ satisfying these conditions and such that the resulting matrix $U$ approximates $R_z(\theta)$ up to $\varepsilon$. The following lemma shows that we can restrict our attention to matrices of determinant 1.

Lemma 4.2. If $\varepsilon < \sqrt{2}$, then all the solutions to Problem 4.1 have the form

\[
U = \frac{1}{\sqrt{5}} \begin{pmatrix}
\alpha & -\beta^* \\
\beta & \alpha^*
\end{pmatrix}, \tag{4}
\]

with $k \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{Z}[i]$. If $\varepsilon \geq \sqrt{2}$, then there exists a solution of $V$-count 0 (i.e., a Pauli operator), and it is also of the form (4).

Proof. It is well-known that every complex $2 \times 2$ unitary operator $U$ can be written as

\[
U = \begin{pmatrix}
a & -b^* e^{i\phi} \\
b & a^* e^{i\phi}
\end{pmatrix},
\]

for $a, b \in \mathbb{C}$ and $\phi \in [-\pi, \pi]$. This, together with the characterization of Pauli+$V$ operators given by Proposition 3.4, implies that a complex $2 \times 2$ unitary operator $U$ can be exactly synthesized over the Pauli+$V$ basis if and only if

\[
U = \frac{1}{\sqrt{5}} \begin{pmatrix}
\alpha & -\beta^*(-1)^\ell \\
\beta & \alpha^*(-1)^\ell
\end{pmatrix},
\]

with $k \in \mathbb{N}$, $\ell \in \{0, 1\}$, and $\alpha, \beta \in \mathbb{Z}[i]$. Now assume that $\varepsilon < \sqrt{2}$ and $||U - R_z(\theta)|| \leq \varepsilon$. Let $e^{i\phi_1}$ and $e^{i\phi_2}$ be the eigenvalues of $UR_z(\theta)^{-1}$, with $\phi_1, \phi_2 \in [-\pi, \pi]$. Then

\[
\sqrt{2} > \varepsilon \geq ||U - R_z(\theta)|| = ||I - UR_z(\theta)^{-1}|| = \max \{|1 - e^{i\phi_1}|, |1 - e^{i\phi_2}|\},
\]

so that $|1 - e^{i\phi_1}| < \sqrt{2}$ and therefore $-\pi/2 < \phi_1 < \pi/2$, for $j \in \{1, 2\}$, which implies that $-\pi < \phi_1 + \phi_2 < \pi$. Hence $|1 - e^{i(\phi_1 + \phi_2)}| < |1 - e^{i\pi}| = 2$. But $e^{i(\phi_1 + \phi_2)} = \det(UR_z(\theta)^{-1}) = (-1)^\ell$. Thus $|1 - (-1)^\ell| < 2$ which proves that $(-1)^\ell = 1$.

For the last statement, note that if $\theta/2 \in [-\pi/2, \pi/2]$, then $||I - R_z(\theta)|| = |1 - e^{i\theta/2}| \leq \sqrt{2}$ and if $\theta/2 \in [\pi/2, 3\pi/2]$, then $||I - R_z(\theta)|| = |1 - e^{i\theta/2}| \leq \sqrt{2}$. Either way $R_z(\theta)$ is approximated to within $\varepsilon$ by a Pauli operator. \hfill \qed

By Lemma 4.2, it therefore suffices to find $k \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{Z}[i]$ such that $\alpha^*\alpha + \beta^*\beta = 5^k$ and such that the resulting $U$ of the form (4) approximates $R_z(\theta)$ up to $\varepsilon$. The key observation here is that, given $\varepsilon$ and $\theta$, we can
express the requirement \( \| U - R_\alpha(\theta) \| \leq \varepsilon \) as a constraint on the top left entry \( \alpha / \sqrt{5^k} \) of \( U \). Indeed, let \( z = e^{-i \theta / 2} \), \( \alpha' = \alpha / \sqrt{5^k} \), and \( \beta' = \beta / \sqrt{5^k} \). Since \( \alpha'^\dagger \alpha' + \beta'^\dagger \beta' = 1 \) and \( z^\dagger z = 1 \), we have

\[
\| R_\alpha(\theta) - U \| = |\alpha' - z|^2 + |\beta'|^2 = (\alpha' - z)^\dagger (\alpha' - z) + \beta'^\dagger \beta' = \alpha'^\dagger \alpha' + \beta'^\dagger \beta' - z^\dagger \alpha' - \alpha'^\dagger z + z^\dagger z = 2 - 2 \text{Re}(z^\dagger \alpha').
\]

Thus \( \| R_\alpha(\theta) - U \| \leq \varepsilon \) if and only if \( 2 - 2 \text{Re}(z^\dagger \alpha') \leq \varepsilon^2 \), or equivalently, \( \text{Re}(z^\dagger \alpha') \geq 1 - \frac{\varepsilon^2}{2} \). If we identify the complex numbers \( z = x + yi \) and \( \alpha' = a + bi \) with 2-dimensional real vectors \( \vec{z} = (x, y)^T \) and \( \vec{\alpha}' = (a, b)^T \), then \( \text{Re}(z^\dagger \alpha') \) is just their inner product \( \vec{z} \cdot \vec{\alpha}' \), and therefore \( \| R_\alpha(\theta) - U \| \) is equivalent to

\[
\vec{z} \cdot \vec{\alpha}' \geq 1 - \frac{\varepsilon^2}{2}.
\tag{5}
\]

Moreover, \( \alpha'^\dagger \alpha' + \beta'^\dagger \beta' = 1 \) implies that \( \alpha'^\dagger \alpha' = 1 - \beta'^\dagger \beta' \leq 1 \) and therefore that \( \vec{\alpha}' \) is an element of the closed unit disk \( \overline{D} \). These two remarks define a subset of the unit disk

\[
\mathcal{R}_\varepsilon = \{ \vec{\alpha}' \in \overline{D} \mid \vec{z} \cdot \vec{\alpha}' \geq 1 - \frac{\varepsilon^2}{2} \},
\tag{6}
\]

which we call the \( \varepsilon \)-region for \( \theta \), such that if \( \alpha' \in \mathcal{R}_\varepsilon \), then \( \| R_\alpha(\theta) - U \| \leq \varepsilon \). In the presence of \( \alpha' = \alpha / \sqrt{5^k} \in \mathcal{R}_\varepsilon \), all that remains is to find the other entry of \( U \) by solving the Diophantine equation

\[
\alpha'^\dagger \alpha + \beta'^\dagger \beta = 5^k
\]

for some unknown \( \beta \in \mathbb{Z}[i] \).

Now recall that we wish to solve Problem 4.1 optimally, so that we need to find an approximating matrix \( U \) whose \( V \)-count is as low as possible. We know from Proposition 3.4 that the \( V \)-count of \( U \) is equal to its least denominator exponent. Therefore if we can enumerate the points of \( \mathcal{R}_\varepsilon \) of the form \( \alpha / \sqrt{5^k} \) for \( \alpha \in \mathbb{Z}[i] \) in order of increasing \( k \), then we can try to solve the Diophantine equation for each such point. In this case, the first candidate for which the Diophantine equation has a solution will yield an optimal solution to Problem 4.1.

Problem 4.1 is therefore equivalent to the following problem.

**Problem 4.3.** Given an angle \( \theta \) and a precision \( \varepsilon > 0 \), find \( k \in \mathbb{N} \) and \( \alpha, \beta \in \mathbb{Z}[i] \) such that:

1. \( \alpha / \sqrt{5^k} \in \mathcal{R}_\varepsilon \),
2. \( \alpha'^\dagger \alpha + \beta'^\dagger \beta = 5^k \),
3. and \( k \) is as small as possible.

In the above problem, the first two goals can be treated as separate problems.

**Problem 4.4** (Scaled grid problem). Given a convex bounded subset \( A \) of \( \mathbb{R}^2 \) with non-empty interior, enumerate all points \( \alpha / \sqrt{5^k} \in A \), where \( \alpha \in \mathbb{Z}[i] \) and \( k \in \mathbb{N} \), in order of increasing \( k \).

**Problem 4.5** (Diophantine equation). Given integers \( \alpha \in \mathbb{Z}[i] \) and \( k \in \mathbb{N} \), find \( \beta \in \mathbb{Z}[i] \) such that \( \alpha'^\dagger \alpha + \beta'^\dagger \beta = 5^k \) if such a \( \beta \) exists.

We now discuss methods to solve both of these problems. We provide an algorithm for Problem 4.1 and analyze its properties in Section 4.3 and Section 4.4 respectively.

### 4.1 Grid problems

In this subsection, we define an efficient algorithm to solve Problem 4.1. In what follows we refer to the set \( \mathbb{Z}^2 \subseteq \mathbb{R}^2 \) as the grid and to elements of \( \mathbb{Z}^2 \) as grid points. The instances of the scaled grid problem where the set \( A \) is an upright rectangle, i.e., of the form \([x_1, x_2] \times [y_1, y_2]\), are easy to solve. If \( A \) is not an upright rectangle, the problem can still be solved efficiently, provided that \( A \) can be “made upright enough”.
**Definition 4.6** (Uprightness). Let $A$ be a bounded convex subset of $\mathbb{R}^2$. The bounding box of $A$, denoted $\text{BBox}(A)$, is the smallest set of the form $[x_1, x_2] \times [y_1, y_2]$ that contains $A$. The *uprightness of $A$*, denoted $\up(E)$, is defined to be the ratio of the area of $A$ to the area of its bounding box:

$$\up(E) = \frac{\text{area}(A)}{\text{area}(\text{BBox}(A))}.$$ 

We say that $A$ is $M$-upright if $\up(E) \geq M$.

We will be especially interested in the case where the set $A$ is an ellipse. Our interest in ellipses is motivated by the fact that a convex bounded subset $A$ of the plane with non-empty interior can always be enclosed in an ellipse whose area differs from that of $A$ by at most a constant factor. To increase the uprightness of a given subset $A$ of the plane, we will then act on its “enclosing ellipse” using linear operators that map the grid to itself.

**Definition 4.7** (Ellipse). Let $D$ be a positive definite real $2 \times 2$-matrix with non-zero determinant, and let $p \in \mathbb{R}^2$ be a point. The *ellipse defined by $D$ and centered at $p$* is the set

$$E = \{u \in \mathbb{R}^2 \mid (u - p)^\top D(u - p) \leq 1\}.$$ 

**Proposition 4.8.** Let $A$ be a bounded convex subset of $\mathbb{R}^2$ with non-empty interior. Then there exists an ellipse $E$ such that $A \subseteq E$, and such that

$$\text{area}(E) \leq \frac{4\pi}{3\sqrt{3}} \text{area}(A).$$

Moreover, $E$ can be efficiently computed.

**Proof.** See theorems 5.17 and 5.18 of [9]. □

The uprightness of an ellipse can be expressed in terms of the entries of its defining matrix. Indeed, let $D$ be the positive definite matrix defining some ellipse $E$ and assume that the entries of $D$ are as follows:

$$D = \begin{pmatrix} a & b \\ b & d \end{pmatrix}.$$ 

We can compute the area of $E$ and the area of its bounding box using $D$: $\text{area}(E) = \pi / \sqrt{\text{det}(D)}$ and $\text{area}(\text{BBox}(E)) = 4\sqrt{ad}/\text{det}(D)$. Thus by Definition 4.6 we get:

$$\up(E) = \frac{\text{area}(E)}{\text{area}(\text{BBox}(E))} = \frac{\pi \sqrt{\text{det}(D)}}{4 \sqrt{ad}}.$$ 

(7)

The uprightness of $E$ is invariant under translation and scalar multiplication.

**Definition 4.9** (Grid operator). A *grid operator* is an integer matrix, or equivalently, a linear operator that maps $\mathbb{Z}^2$ to itself. A grid operator $G$ is called *special* if it has determinant $\pm 1$, in which case $G^{-1}$ is also a grid operator.

**Remark 4.10.** If $A$ is a subset of $\mathbb{R}^2$ and $G$ is a grid operator, then $G(A)$, the direct image of $A$, is defined as usual by $G(A) = \{G(v) \mid v \in A\}$. If $G$ is a grid operator and $E$ is an ellipse centered at the origin and defined by $D$, then $G(E)$ is an ellipse defined by $(G^{-1})^\top DG^{-1}$.

**Proposition 4.11.** Let $E$ be an ellipse, defined by $D$ and centered at $p$. There exists a grid operator $G$ such that $G(E)$ is $1/2$-upright. Moreover, if $E$ is $M$-upright, then $G$ can be efficiently computed in $O(\log(1/M))$ arithmetic operations.

**Proof.** If $E$ is an ellipse defined by a matrix $D$, we write $\text{Skew}(E)$ for the product of the anti-diagonal entries of $D$. Let $A$ and $B$ be the following special grid operators:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

and consider an arbitrary ellipse $E$. Since uprightness is invariant under translation and scaling, we may without loss of generality assume that $E$ is centered at the origin and that $D$ has determinant 1. Suppose moreover that the entries of $D$ are as follows:

$$\begin{pmatrix} a & b \\ b & d \end{pmatrix}$$

7
We first show that there exists a grid operator $G$ such that $\text{Skew}(G(E)) \leq 1$. Indeed, assume that $\text{Skew}(E) = b^2 \geq 1$. In case $a \leq d$, choose $n$ such that $|na + b| \leq a/2$. Then we have:

$$A^nDA^n = \begin{pmatrix} \cdots & na + b \\ na + b & \cdots \end{pmatrix}. $$

Therefore, using Remark 4.10 with $G_1 = (A^n)^{-1}$, we have:

$$\text{Skew}(G_1(E)) = (na + b)^2 \leq \frac{a^2}{4} \leq \frac{ad}{4} = \frac{1 + b^2}{4} = \frac{1 + \text{Skew}(E)}{4} \leq \frac{2\text{Skew}(E)}{4} = \frac{1}{2}\text{Skew}(E).$$

Similarly, in case $d < a$, then choose $n$ such that $|na + b| \leq d/2$. A similar calculation shows that in this case, with $G_1 = (B^n)^{-1}$, we get $\text{Skew}(G_1(E)) \leq \frac{1}{2}\text{Skew}(E)$. In both cases, the skew of $E$ is reduced by a factor of 2 or more.

Now let $D'$ be the matrix defining $G(E)$, with entries as follows:

$$D' = \begin{pmatrix} \alpha & \beta \\ \beta & \delta \end{pmatrix}. $$

Then $\text{Skew}(G(E)) \leq 1$ implies that $\beta^2 \leq 1$. Moreover, since $A$ and $B$ are special grid operators we have $\det(D') = \alpha\delta - \beta^2 = 1$. Using the expression (4) for the uprightness of $G(E)$ we get the desired result:

$$\text{up}(G(E)) = \frac{\pi}{4} \sqrt{\frac{\det(D')}{\alpha\delta}} = \frac{\pi}{4\sqrt{\alpha\delta}} = \frac{\pi}{4\sqrt{\beta^2 + 1}} \geq \frac{\pi}{4\sqrt{2}} \geq \frac{1}{2}. $$

Finally, to bound the number of arithmetic operations, note that each application of $G_j$ reduces the skew by at least a factor of 2. Therefore, the number $n$ of grid operators required satisfies $n \leq \log_2(\text{Skew}(E))$. Now note that since $D$ has determinant 1, we have:

$$M \leq \text{up}(E) = \frac{\pi}{4} \frac{1}{\sqrt{ad}} = \frac{\pi}{4\sqrt{b^2 + 1}}. $$

Therefore $\text{Skew}(E) = b^2 \leq (\pi^2/16M^2) - 1$, so that the computation of $G$ requires $O(\log(1/M))$ arithmetic operations.

We can now describe our algorithm to solve Problem 4.4.

**Proposition 4.12.** There is an algorithm which, given a bounded convex subset $A$ of $\mathbb{R}^2$ with non-empty interior, enumerates all solutions of the grid problem for $A$ in order of increasing $k$. Moreover, if $A$ is $M$-upright, then the algorithm requires $O(\log(1/M))$ arithmetic operations overall, plus a constant number of arithmetic operations per solution produced.

**Proof.** Given $A$, start by using Proposition 4.8 to find an ellipse $A'$ containing $A$ and whose area only exceeds that of $A$ by a fixed constant factor $N$. Next, use Proposition 4.11 to find a grid operator $G$ such that $G(A')$ is 1/2-upright. Now enumerate the grid points of $\text{BBox}(G(A'))$ in order of increasing $k$. This can be done efficiently since $\text{BBox}(G(A'))$ is an upright rectangle. For each grid point $u$ found, check whether it belongs to $G(A)$. This is the case if and only if $G^{-1}(u)$ is a solution to the grid problem for $A$ with denominator exponent $k$. 

**4.2 Diophantine equations**

There is a well-known algorithm to solve Problem 4.8, i.e., to solve the equation:

$$\alpha^1\alpha + \beta^1\beta = 5^k, \quad (8)$$

where $\alpha, \beta \in \mathbb{Z}[i]$, and $k \in \mathbb{N}$. First note that if we write $n = 5^k - \alpha^1\alpha$ and $\beta = b + ic$, where $n, b, c \in \mathbb{Z}$, then Equation (8) is equivalent to

$$n = b^2 + c^2. \quad (9)$$

The solutions to Equation (9) were characterized by Euler:
Proposition 4.13 (Euler [2]). Let n be a positive integer with prime factorization \( p_1^{k_1} \ldots p_m^{k_m} \), where \( p_1, \ldots, p_m \) are distinct positive primes. Then n can be written as the sum of two squares if and only if for all i either \( k_i \) is even or \( p_i \equiv 1, 2 \pmod{4} \).

**Proof.** See Theorem 366 of [1].

Moreover, in case the equation \( n = b^2 + c^2 \) has a solution, there is an efficient probabilistic algorithm for finding \( b \) and \( c \), given a prime factorization for \( n \), see [8].

4.3 The approximate synthesis algorithm

We can now describe our algorithm to solve Problem 4.1.

**Algorithm 4.14.** Given \( \theta \) and \( \varepsilon \), let \( A = R_\varepsilon \) be the \( \varepsilon \)-region as defined in Equation (6).

1. Use Proposition 4.12 to enumerate the infinite sequence of solutions \( \alpha \) to the scaled grid problem for \( A \) and \( k \) in order of increasing least denominator exponent \( k \).

2. For each such solution \( \alpha \) of least denominator exponent \( k \):
   (a) Let \( n = 5^k(1 - \alpha^* \alpha) \).
   (b) Attempt to find a prime factorization of \( n \). If \( n \neq 0 \) but no prime factorization is found, skip step 2(c) and continue with the next \( \alpha \).
   (c) Use the algorithm of Section 4.2 to solve the equation \( \beta^* \beta = n \). If a solution \( \beta \) exists, go to step 3; otherwise, continue with the next \( \alpha \).

3. Define \( U \) as in Equation (4) and use the exact synthesis algorithm of Proposition 3.4 to find a Pauli+V circuit implementing \( U \). Output this circuit and stop.

4.4 Analysis of the algorithm

4.4.1 Correctness

**Proposition 4.15.** If Algorithm 4.14 terminates, then it yields a valid solution to the approximate synthesis problem, i.e., it yields a Pauli+V circuit approximating \( R_\varepsilon(\theta) \) up to \( \varepsilon \).

**Proof.** By construction, following the reduction of Problem 4.1 to Problem 4.3.

4.4.2 Optimality in the presence of a factoring oracle

**Proposition 4.16.** In the presence of an oracle for integer factoring, the circuit returned by Algorithm 4.14 has the smallest \( V \)-count of any single-qubit Pauli+V circuit approximating \( R_\varepsilon(\theta) \) up to \( \varepsilon \).

**Proof.** By construction, step 1 of the algorithm enumerates all solutions \( \alpha \) to the scaled grid problem for \( R_\varepsilon \) in order of increasing denominator exponent \( k \). Step 2(a) always succeeds and, in the presence of the factoring oracle, so does step 2(b). When step 2(c) succeeds, the algorithm has found a solution of Problem 4.3 for a minimal \( k \).

4.4.3 Near-optimality in the absence of a factoring oracle

The proof that our algorithm is nearly optimal in the absence of a factoring oracle relies on the following number-theoretic hypothesis. We do not have a proof of this hypothesis, but it appears to be valid in practice.

**Hypothesis 4.17.** For each number \( n \) produced in step 2(a) of Algorithm 4.14, write \( n = 2^l m \), where \( m \) is odd. Then \( m \) is asymptotically as likely to be a prime congruent to 1 modulo 4 as a randomly chosen odd number of comparable size. Moreover, each \( m \) can be modelled as an independent random variable.

**Lemma 4.18.** Let \( A \) be a convex subset of \( \mathbb{R}^2 \), \( k \geq 0 \), and assume that the two-dimensional scaled grid problem for \( A \) has at least two distinct solutions with denominator exponent \( k \). Then for all \( \ell \geq 0 \), the scaled grid problem for \( A \) has at least \( 5^\ell + 1 \) solutions with denominator exponent \( k + 2\ell \).
Proof. Let \( \alpha \neq \beta \) be solutions of the scaled grid problem for \( A \) with denominator exponent \( k \). For each \( j = 0, 1, \ldots, 5^\ell \), let \( \phi = \frac{1}{\beta} \), and consider \( \alpha_j = \phi \alpha + (1 - \phi) \beta \). Then \( \alpha_j \) has denominator exponent \( k + 2\ell \). Also, \( \alpha_j \) is a convex combination of \( u \) and \( v \). Since \( A \) is convex, it follows that \( \alpha_j \) is a solution of the scaled grid problem for \( A \), yielding \( 5^\ell + 1 \) distinct solutions with denominator exponent \( k + 2\ell \). \hfill \Box

Lemma 4.19. Let \( b > 0 \) be an arbitrary fixed constant. Then for \( a \geq 1 \),

\[
\sum_{x=1}^{\infty} \left( 1 - \frac{1}{a + b \ln x} \right)^x = O(a).
\]

Proof. The lemma is proved in Appendix E of [9]. \hfill \Box

Definition 4.20. Let \( U' \) and \( U'' \) be two solutions of the approximate synthesis problem of the form

\[
U' = \left( \frac{\alpha'}{\beta'} - \frac{\beta'}{\alpha'}^a \right) \quad \text{and} \quad U'' = \left( \frac{\alpha''}{\beta''} - \frac{\beta''}{\alpha''}^a \right).
\]

We say that \( U' \) and \( U'' \) are equivalent solutions if \( \alpha' = \alpha'' \).

Proposition 4.21. Let \( k \) be the V-count of the solution of the approximate synthesis problem found by Algorithm 4.14 in the absence of a factoring oracle. Then

1. The approximate synthesis problem has at most \( O(\log(1/\varepsilon)) \) non-equivalent solutions with V-count less than \( k \).

2. The expected value of \( k \) is \( k'' + O(\log(1/\varepsilon)) \), where \( k', k'', \) and \( k''' \) are the V-counts of the optimal second-to-optimal and third-to-optimal solutions of the approximate synthesis problem (up to equivalence).

Proof. If \( \varepsilon \geq \sqrt{2} \), then by Lemma 4.12 there is a solution of V-count 0 and the algorithm easily finds it. In this case there is nothing to show, so assume without loss of generality that \( \varepsilon < \sqrt{2} \). Then by Lemma 4.12, all solutions are of the form (1).

1. Consider the list \( \alpha_1, \alpha_2, \ldots \) of candidates generated in step 1 of the algorithm. Let \( k_1, k_2, \ldots \) be their least denominator exponent and let \( n_1, n_2, \ldots \) be the corresponding integers calculated in step 2(a). Note that \( n_j \leq 5^{k_j} \) for all \( j \). Write \( n_j = 2^{j_1} m_j \) where \( m_j \) is odd. By Hypothesis 4.17 the probability that \( m_j \) is a prime congruent to 1 modulo 4 is asymptotically no smaller than that of a randomly chosen odd integer less than \( 5^{k_j} \), which, by the well-known prime number theorem, is

\[
p_j := \frac{1}{\ln(5^{k_j})} = \frac{1}{k_j \ln 5}.
\]

By the pigeon-hole principle, two of \( k_1, k_2, \) and \( k_3 \) must be congruent modulo 2. Assume without loss of generality that \( k_2 \equiv k_3 \mod 2 \). Then \( \alpha_2 \) and \( \alpha_3 \) are two distinct solutions to the scaled grid problem for \( R \) with (not necessarily least) denominator exponent \( k_3 \). It follows by Lemma 4.18 that there are at least \( 5^\ell + 1 \) distinct candidates of denominator exponent \( k_3 + 2\ell \), for all \( \ell \geq 0 \). In other words, for all \( j \), if \( j \leq 5^\ell + 1 \), we have \( k_j \leq k_3 + 2\ell \). In particular, this holds for \( \ell = \lfloor \log_5 j \rfloor \), and therefore,

\[
k_j \leq k_3 + 2(1 + \log_5 j).
\]

Combining (12) with (11), we have

\[
p_j \geq \frac{1}{(k_3 + 2(1 + \log_5 j)) \ln 5} = \frac{1}{(k_3 + 2) \ln 5 + 2 \ln j}.
\]

Let \( j_0 \) be the smallest index such that \( m_{j_0} \) is a prime congruent to 1 modulo 4. By Hypothesis 4.17 we can treat each \( m_j \) as an independent random variable. Therefore,

\[
P(j_0 > j) = P(m_1, \ldots, n_j \text{ are not prime}) \\
\leq (1 - p_1)(1 - p_2) \cdots (1 - p_j) \\
\leq (1 - p_j)^j \\
\leq \left( 1 - \frac{1}{(k_3 + 2) \ln 5 + 2 \ln j} \right)^j.
\]
The expected value of \( j_0 \) is
\[
E(j_0) = \sum_{j=0}^{\infty} P(j_0 > j) \leq 1 + \sum_{j=1}^{\infty} \left(1 - \frac{1}{(k_3+2) \ln 5 + 2 \ln j}\right)^j \approx O(k_3),
\]
where we have used Lemma 4.19 to estimate the sum.

Next, we will estimate \( k_3 \). First note that if the \( \varepsilon \) region contains a circle of radius greater than \( 1/\sqrt{5}^5 \), then it contains at least 3 solutions to the scaled grid problem for \( R_\varepsilon \) with denominator exponent \( k \). The width of the \( \varepsilon \)-region \( R_\varepsilon \) is \( \varepsilon^2/2 \) at the widest point, and we can inscribe a disk of radius \( r = \varepsilon^2/4 \) in it. Hence the scaled grid problem for \( R_\varepsilon \), as in step 1 of the algorithm, has at least three solutions, provided that
\[
r = \frac{\varepsilon^2}{4} \geq \frac{1}{\sqrt{5}^k},
\]
or equivalently, that
\[
k \geq 2 \log_5(2) + 2 \log_5(1/\varepsilon).
\]
It follows that
\[
k_3 = O(\log(1/\varepsilon)),
\]
and therefore, using (14), also
\[
E(j_0) = O(\log(1/\varepsilon)).
\]

To finish the proof of part (a), recall that \( j_0 \) was defined to be the smallest index such that \( m_{j_0} \) is a prime congruent to 1 modulo 4. The primality of \( m_{j_0} \) ensures that step 2(b) of the algorithm succeeds for the candidate \( \alpha_{j_0} \). Furthermore, because \( m_{j_0} \equiv 1 \pmod{4} \), the equation \( \beta^j \beta = n \) has a solution by Proposition 4.13. Hence the remaining steps of the algorithm also succeed for \( u_{j_0} \).

Now let \( r \) be the number of non-equivalent solutions of the approximate synthesis problem of \( V \)-count strictly less than \( k \). As noted above, any such solution \( U \) is of the form (4). Then the least denominator exponent of \( \alpha \) is strictly smaller than \( k_{j_0} \), so that \( \alpha = \alpha_j \) for some \( j < j_0 \). In this way, each of the \( r \) non-equivalent solutions is mapped to a different index \( j < j_0 \). It follows that \( r < j_0 \), and hence \( E(r) \leq E(j_0) = O(\log(1/\varepsilon)) \), as was to be shown.

2. Let \( U' \) be an optimal solution of the exact synthesis problem, let \( U'' \) be optimal among the solutions that are not equivalent to \( U' \) and let \( U''' \) be optimal among the solutions that are not equivalent to either \( U' \) or \( U'' \). Assume that \( U', U'', \) and \( U''' \) are written as in (10) with top-left entry \( \alpha', \alpha'', \) and \( \alpha''' \) respectively. Now let \( k', k'', \) and \( k''' \) be the least denominator exponents of \( \alpha', \alpha'' \) and \( \alpha''' \), respectively. Let \( k_3 \) and \( j_0 \) be as in the proof of part 1. Note that, by definition, \( k_3 \leq k'' \). Let \( k \) be the least denominator exponent of the solution of the approximate synthesis problem found by the algorithm. Then \( k \leq k_{j_0} \). Using (12), we have
\[
k \leq k_{j_0} \leq k_3 + 2(1 + \log_5 j_0) \leq k''' + 2(1 + \log_2 j_0).
\]
This calculation applies to any one run of the algorithm. Taking expected values over many randomized runs, we therefore have
\[
E(k) \leq k''' + 2 + 2E(\log_5 j_0) \leq k''' + 2 + 2\log_5 E(j_0).
\]
Note that we have used the law \( E(\log j_0) \leq \log(E(j_0)) \), which holds because \( \log \) is a concave function. Combining (17) with (16), we therefore have the desired result:
\[
E(k) = k''' + O(\log(\log(1/\varepsilon))).
\]

\[\Box\]

### 4.4.4 Time complexity

**Proposition 4.22.** Algorithm 4.14 runs in expected time \( O(\text{polylog}(1/\varepsilon)) \). This is true whether or not a factorization oracle is used.

**Proof.** This proposition is proved like the corresponding one in [9].

\[\Box\]
5 Conclusion

We have introduced an algorithm for the approximate synthesis of Pauli+V circuits. Our algorithm is optimal if an oracle for the factorization of integers is available. In the absence of such an oracle, our algorithm is still nearly optimal, yielding circuits of V-count \( m + O(\log(\log(1/\varepsilon))) \), where \( m \) is the V-count of the third-to-optimal solution. To the author’s knowledge, this is the first optimal synthesis algorithm for the Pauli+V basis.

Acknowledgements

The author would like to thank Peter Selinger for his helpful comments.

References

[1] A. Bocharov, Y. Gurevich, and K. M. Svore. Efficient decomposition of single-qubit gates into V basis circuits. *Phys. Rev. A*, 88:012313 (13 pages), 2013. Also available from [arXiv:1303.1411](http://arxiv.org/abs/1303.1411).

[2] L. Euler. De numeris, qui sunt aggregata duorum quadratorum. *Novi Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 4:3–40, 1758.

[3] B. Giles and P. Selinger. Exact synthesis of multiqubit Clifford+T circuits. *Physical Review A*, 87:032332, 2013. Also available from [arXiv:1212.0506](http://arxiv.org/abs/1212.0506).

[4] G. Hardy and E. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 6th edition, 2008.

[5] A. Harrow, B. Recht, and I. Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43, 2002.

[6] A. Lubotzky, R. Phillips, and P. Sarnak. Hecke operators and distributing points on the sphere I. *Communications on Pure and Applied Mathematics*, 39:S149–S186, 1986.

[7] A. Lubotzky, R. Phillips, and P. Sarnak. Hecke operators and distributing points on \( S^2 \). *Communications on Pure and Applied Mathematics*, 40:401–420, 1987.

[8] M. O. Rabin and J. O. Shallit. Randomized algorithms in number theory. *Communications on Pure and Applied Mathematics*, 39:S239–S256, 1986.

[9] N. J. Ross and P. Selinger. Optimal ancilla-free Clifford+T approximation of z-rotations. [arXiv:1403.2975](http://arxiv.org/abs/1403.2975), Mar. 2014.

[10] P. Selinger. Efficient Clifford+T approximation of single-qubit operators. *Quantum Information and Computation*, 2014. To appear. Also available from [arXiv:1212.6253](http://arxiv.org/abs/1212.6253).

[11] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. Also available from [arXiv:quant-ph/9508027](http://arxiv.org/abs/quant-ph/9508027).