Quantum Information and Computation, Vol. 23, No. 11&12 (2023) 0901–0923
© Rinton Press

FAST NAVIGATION WITH ICOSAHEDRAL GOLDEN GATES

TERRENCE RICHARD BLACKMAN
Department of Mathematics, Medgar Evers College
Brooklyn, New York 11225, United States of America

ZACHARY STIER
Department of Mathematics, University of California, Berkeley
Berkeley, California 94720, United States of America

Received 14 November 2022
Revised 25 July 2023

An algorithm of Ross and Selinger for the factorization of diagonal elements of PU(2) to within distance ε was adapted by Parzanchevski and Sarnak into an efficient probabilistic algorithm for any element of PU(2) using at most effective $3\log_p \frac{1}{\varepsilon^3}$ factors from certain well-chosen sets associated to a number field and a prime p. The icosahedral super golden gates are one such set associated to $\mathbb{Q}(\sqrt{5})$. We leverage recent work of Carvalho Pinto, Petit, and Stier to reduce this bound to $\frac{7}{3}\log_{59}\frac{1}{\varepsilon^3}$, and we implement the algorithm in Python. This represents an improvement by a multiplicative factor of $\log_2 59 \approx 5.9$ over the analogous result for the Clifford+T gates. This is of interest because the icosahedral gates have shortest factorization lengths among all super golden gates.

Keywords: Quantum computing, quaternion algebras

1 Introduction

Lubotzky, Phillips, and Sarnak, in a series of papers [1, 2, 3, 4], explicitly constructed topological generators with optimal covering properties for the compact Lie group PU(2). Such generators for projective unitary groups find an interesting application in quantum computing where a fundamental design challenge is to determine an optimal, fault-tolerant decomposition of a quantum gate. For classical computing a single bit state is an element of $\{0, 1\}$. A classical gate implements functions on binary inputs. The only nontrivial single bit logic operation is `NOT`, which takes 0 to 1 and 1 to 0 (though it is also possible for the codomain to contain more than one bit). In the quantum setting, single *qubit* states are points

$$u = (u_1, u_2) \in \mathbb{C}^2$$

up to a mutual phase $e^{2\pi i\theta}$ in each component, such that

$$|u|^2 = |u_1|^2 + |u_2|^2 = 1.$$

A gate here *cannot* output more than one qubit, and thus must be a 2×2 projective unitary.

A *universal gate set* is a finite set of gates, $S := \{s_1, s_2, \cdots, s_k : s_\ell \in \mathrm{PU}(2)\}$, that can approximate, in the bi-invariant metric on the compact Lie group, any matrix arbitrarily well. That is, the group generated by S must be topologically dense in PU(2). The Solovay–Kitaev theorem [5] guarantees that universal gate sets can efficiently approximate quantum operations for unitaries on a constant number of qubits. Universal gate sets typically consists of a finite group $C \leq \mathrm{PU}(2)$ together with an extra element τ, which we will take to be an involution, so that the subgroup generated by C and τ covers PU(2) with minimal τ-count, and simultaneously navigates PU(2) efficiently. That is, given some gate in PU(2) and desired precision ε, there is an efficient algorithm (polynomial-time in the input size) that with high probability finds a short word in S to that precision, typically of length $O(\log(1/\varepsilon))$. The deep insight of Sarnak [6] is that the construction and optimality of universal single-qubit quantum gate sets can be understood in terms of the arithmetic of quaternion algebras. Specifically, identifying C with a subgroup of the group of units in a definite quaternion algebra over a totally real number field provides a coherent framework within which one can systematically address the question of optimality of topological generators for PU(2).

The question we address within this context is that of finding the "best" topological generators of PU(2) among those universal gate sets, the *super golden gates* of Parzanchevski and Sarnak [7], which are known to possess optimal covering properties and efficient navigation. In particular, there are only finitely many super golden gates [7, p.870–871] and the respective finite subgroups C can be realized as the group of symmetries of of the Platonic solids: for the tetrahedron, A_4; for the cube and the octahedron, S_4; and for the dodecahedron and icosahedron, A_5.

We demonstrate that the *icosahedral* super golden gates admit a factorization directly analogous to the one obtained by Stier [8], and that this gives the best-known preconstant to the first-power logarithm in the approximation length. The exact factor of the improvement is $\log_2 59 \approx 5.9$. This improvement is due to the fact that the icosahedral super golden gates have a growth rate that is on the order of 59^k, while gate set studied in [8], the Clifford+T gates, have growth rate of order 2^k. We note also that the *icosahedral* super-golden-gates represent the greatest number of distinct gates with bounded τ-count.

The commonly used Clifford+T gate set provides a set of elementary gates that is universal and consists only of a small number of gates, all of which are very well compatible with many established error correction schemes and can be physically implemented in all quantum technologies that seem promising for large-scale quantum computations [9]. In this case the finite group C is the Clifford group C_{24} of order 24 in PU(2). At least one non-Clifford gate must be added to the basic gate set in order to achieve universality. A common choice for this additional gate is the T-gate (or $\frac{\pi}{8}$-gate). The T-gate is not the only possible extension of the Clifford group but it is considered to be the most practical one. This is due to the availability of fault-tolerant implementations of the T-gate. For this reason, the Clifford+T gate set is considered the most promising candidate for practical quantum computing. We recall, for the convenience of the reader, some of the salient details of the single qubit Clifford+T gate set. Let

$$H := \frac{i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } T := \begin{pmatrix} e^{i\pi/8} & \\ & e^{-i\pi/8} \end{pmatrix}.$$

We take $S := \{H, T\}$ [6, 8, 10]. The Clifford+T universal gate set is an example of a golden gate set [6, 7]. The remarkable observation of [6] is that the H and T gates of the Clifford+T

gate set come from the definite quaternion algebra

$$\mathcal{A} := \left(\frac{-1,-1}{\mathbb{Q}(\sqrt{2})}\right).$$

Kliuchnikov, Maslov, and Mosca [11] characterized the group $\langle S\rangle$ and demonstrated an efficient algorithm to factor exactly, by carefully studying the powers of $\sqrt{2}$ in the denominators of matrix entries. Ross and Selinger [10] then focused on diagonal matrices near to $\langle S\rangle$, with the goal of finding ε-close factorizations of length $c\log_2({}^1\!/\!\varepsilon^3)$; the base of 2 is intrinsic to the structure of the Clifford+T gates. By studying *upright sets* in the plane, which function analogously to the simplices of Lenstra's algorithm [12, 13], they achieved the leading coefficient of $1+o(1)$ (in that restricted case). Parzenchevski and Sarnak [7] generalized Ross and Selinger's approach to any golden gate set (with the base of the logarithm correspondingly changing per the gate set), and by considering Euler angles reached the coefficient of $3+o(1)$ for approximations to generic elements. Working with the (finite) LPS Ramanujan graphs (see §2.1.3 and [4]), Carvalho Pinto and Petit [14] factorize in the equivalent of ${}^7\!/\!3+o(1)$. This approach is adapted by Stier [8] for the same coefficient with Clifford+T gates. We combine aspects of the techniques of [14, 8], in the proposed icosahedral setting, to reduce this bound to $\frac{7}{3}\log_{59}({}^1\!/\!\varepsilon^3)$.

2 Super Golden Gates

We recall here essential ideas related to the arithmetic of quaternion algebras [15, 16], S-arithmetic groups [17, §C], and Ramanujan graphs [18] which lead to the icosahedral super golden gates [7].

2.1 Quaternion algebras

A quaternion algebra $\mathcal{A}$ over a field $\mathbb{F}$ is a central simple algebra of dimension four over $\mathbb{F}$ (cf. [19, §5.2] or [20, p.15]). It follows from Wedderburn's structure theorem on simple algebras [21] that every quaternion algebra over any field $\mathbb{F}$ is either isomorphic to $\mathrm{M}_2(\mathbb{F})$ or a division algebra with center $\mathbb{F}$ [22]. If the characteristic of $\mathbb{F}$ is not 2 then it is always possible to find a basis $\{1,i,j,k\}$ for $\mathcal{A}$ over $\mathbb{F}$ such that

$$i^2=\alpha, j^2=\beta, k=ij=-ji$$

where $\alpha,\beta\in\mathbb{F}^\times$. We designate such an algebra by

$$\mathcal{A}=\left(\frac{\alpha,\beta}{\mathbb{F}}\right).$$

Evidently $q\in\mathcal{A}$ is of the form

$$q=x_0+x_1i+x_2j+x_3k, \tag{1}$$

where $x_\ell\in\mathbb{F}$. In this notation, Hamilton's quaternions arise as

$$\mathbb{H}=\left(\frac{-1,-1}{\mathbb{R}}\right).$$

The conjugate of q as in (1), denoted by $\bar{q}$, is equal to $x_0 - x_1 i - x_2 j - x_3 k$. For each $q \in \mathcal{A}$ we define the (reduced) norm map $\mathrm{N} : \mathcal{A} \to \mathbb{F}$ by

$$\mathrm{N}(q) := q\bar{q} = x_0^2 - \alpha x_1^2 - \beta x_2^2 + \alpha\beta x_3^2$$

and the (reduced) trace map $\mathrm{Tr} : \mathcal{A} \to \mathbb{F}$ by

$$\mathrm{Tr}(q) := q + \bar{q} = 2x_0.$$

We note that every element $q \in \mathcal{A}$ satisfies the quadratic equation

$$q^2 - \mathrm{Tr}(q)q + \mathrm{N}(q) = 0.$$

It is possible to make everything completely explicit by embedding $\mathcal{A}$ in $\mathrm{M}_2(F(\sqrt{\alpha}))$ by, for example,

$$i \mapsto \begin{pmatrix} \sqrt{\alpha} & \\ & -\sqrt{\alpha} \end{pmatrix} \text{ and } j \mapsto \begin{pmatrix} & \beta \\ 1 & \end{pmatrix}.$$

Evidently, if α is a square in $\mathbb{F}$, then $\mathcal{A} \cong \mathrm{M}_2(\mathbb{F})$. A necessary and sufficient condition for $\mathcal{A} \cong \mathrm{M}_2(\mathbb{F})$ is that α is the norm of an element in $\mathbb{F}(\sqrt{\beta})$ with respect to $\mathbb{F}$ [16]. Of course, one may interchange α and β in this remark.

Specializing to quaternion algebras over the rational field $\mathbb{Q}$, or over one of the completions $\mathbb{Q}_p$ or $\mathbb{R}$ (the completion $\mathbb{Q}_\infty$), let $\mathcal{A}$ be a quaternion algebra over $\mathbb{Q}$ and let p be a prime (or ∞). We define

$$\mathcal{A}_p := \mathcal{A} \underset{\mathbb{Q}}{\otimes} \mathbb{Q}_p.$$

Either $\mathcal{A}_p \cong \mathrm{M}_2(\mathbb{Q}_p)$ or $\mathcal{A}_p$ is a division algebra over $\mathbb{Q}_p$. In this case we will have that $\mathcal{A}_p \cong \mathbb{H}_p$ and say that $\mathcal{A}$ is *ramified* at p. When $\mathcal{A}_p \cong \mathrm{M}_2(\mathbb{Q}_p)$ we will say that $\mathcal{A}$ is *unramified* or *split* at p. If $\mathcal{A}$ is ramified at ∞ it is called a *definite rational quaternion algebra*—that is, if $\mathcal{A}$ is definite then

$$\mathcal{A}_\infty = \mathcal{A} \underset{\mathbb{Q}}{\otimes} \mathbb{R} \cong \mathbb{H}.$$

If $\mathcal{A}$ is split at ∞, it is called an *indefinite rational quaternion algebra*—that is, if $\mathcal{A}$ is indefinite then

$$\mathcal{A}_\infty = \mathcal{A} \underset{\mathbb{Q}}{\otimes} \mathbb{R} \cong \mathrm{M}_2(\mathbb{R}).$$

2.1.1 Definite quaternion algebras over totally real number fields

We now turn to the quaternion algebras that are the objects of interest in this paper, definite quaternion algebras over totally real number fields. Let $\mathcal{A}$ be a quaternion algebra over a number field $\mathbb{F}$, ν be a place of $\mathbb{F}$, and $\mathbb{F}_\nu$ be the completion of $\mathbb{F}$ at ν. Recall that a *totally real number field* is a finite algebraic extension of $\mathbb{Q}$ all of whose complex embeddings lie entirely in $\mathbb{R}$. For example, the field $\mathbb{Q}(\sqrt{d})$ is totally real for positive, integral d. If every infinite place of F is ramified in $\mathcal{A}$, we say that $\mathcal{A}$ is a totally definite quaternion algebra. Consequently, if $\mathcal{A}$ is a totally definite quaternion algebra over a number field $\mathbb{F}$, then $\mathbb{F}$ is necessarily totally real. Moreover, if $\mathbb{F}$ is a quadratic field, the number of finite places which are ramified in $\mathcal{A}$ is even.

2.1.2 Unit groups in orders in definite quaternion algebras over totally real number fields

Let $\mathcal{A}$ be quaternion algebra over a field $\mathbb{F}$ and let $O_{\mathbb{F}}$ be the ring of integers in $\mathbb{F}$. An element $q \in \mathcal{A}$ is *integral* if $\mathrm{N}(q), \mathrm{Tr}(q) \in O_{\mathbb{F}}$. An *order* $\mathcal{O} \subseteq \mathcal{A}$ is a $O_{\mathbb{F}}$-algebra of integral elements such that [23, p.2]

$$\mathcal{O} \underset{O_{\mathbb{F}}}{\otimes} \mathbb{F} \cong \mathcal{A}.$$

Observe that as an example of an order we always have

$$\mathcal{O} := O_{\mathbb{F}} \oplus O_{\mathbb{F}} i \oplus O_{\mathbb{F}} j \oplus O_{\mathbb{F}} k. \tag{2}$$

If $\mathcal{O}$ is an $O_{\mathbb{F}}$-order in a definite quaternion algebra $\mathcal{A}$ over the totally real field $\mathbb{F}$ then the group of units of reduced norm 1, i.e.

$$\mathcal{O}^1 := \{\alpha \in \mathcal{O} : \mathrm{N}(\alpha) = 1\} \tag{3}$$

is a finite group [16, p.289]. These finite groups will correspond to the groups of rotational symmetries of the platonic solids [16, p.172–173].

2.1.3 Ramanujan graphs

Two key ideas are needed for the construction of golden gate and super golden gate sets: a S-arithmetic unit quaternion group and a Ramanujan graph. We outline the essential ideas of Lubotzky, Phillips, and Sarnak's [4] "LPS" construction of these objects.

Ramanujan graphs are graphs whose spectrum is bounded optimally. Let X be a finite connected k-regular graph and A its adjacency matrix.

Definition 1. *The graph X is called a Ramanujan graph if every eigenvalue λ of A satisfies either $|\lambda| = k$ or $|\lambda| \leq 2\sqrt{k-1}$.*

LPS Ramanujan graphs [24] arise as Cayley graphs of $\mathrm{PSL}(2, \mathbb{F}_q)$ (for $\mathbb{F}_q$ the finite field on q elements). When considering these graphs we interchangeably refer to their elements by their group-theoretic properties as matrices and their graph-theoretic relations as vertices. [4] establishes that for any prime $p \equiv 1 \pmod 4$ there are infinitely many $(p+1)$-regular Ramanujan graphs. We use the notation $X^{p,q}$ where p and q are distinct primes congruent to 1 modulo 4 to represent such graphs. The construction comes from number theory by way of the generalized Ramanujan conjecture [7, p.873]. The symmetric space $\mathrm{PGL}(2, \mathbb{Q}_p)/\mathrm{PGL}(2, \mathbb{Z}_p)$ can be identified with a $(p+1)$-regular infinite tree. $\mathrm{PGL}(2, \mathbb{Z}[1/p])$ acts from the left on $\mathrm{PGL}(2, \mathbb{Q}_p)/\mathrm{PGL}(2, \mathbb{Z}_p)$. The generalized Ramanujan conjecture, a theorem in this case, implies that the quotient of $\mathrm{PGL}(2, \mathbb{Q}_p)/\mathrm{PGL}(2, \mathbb{Z}_p)$ by any congruence subgroup of $\mathrm{PGL}(2, \mathbb{Z}[1/p])$, a $(p+1)$-regular graph, is a Ramanujan graph. By considering an appropriate congruence subgroup of $\mathrm{PGL}(2, \mathbb{Z}[1/p])$ we can identify the quotient of this symmetric space with a Cayley graph associated to $\mathrm{PSL}(2, \mathbb{F}_q)$ or $\mathrm{PGL}(2, \mathbb{F}_q)$, depending on the value of the Legendre symbol $\left(\frac{p}{q}\right)$ [25].

2.1.4 p-arithmetic unit quaternion groups

Golden gate and super golden gate sets for PU(2) require the construction of a p-arithmetic group (a special case of S-arithmetic groups, where S is a collection of places of $\mathbb{Z}$). Let $G \leq \mathrm{GL}(n)$ be an algebraic group defined over $\mathbb{Z}[1/p]$ with $G(\mathbb{R})$ compact. A p-arithmetic group Λ is a subgroup of $G(\mathbb{Z}[1/p]) \leq G(\mathbb{R}) \times G(\mathbb{Q}_p)$, and has *congruence subgroup* $\Lambda(N) :=$

$\{g \in \Lambda : g \equiv I \pmod{N}\}$. That is, in our compact Lie group PU(2) we take only rational numbers whose denominators are powers of a fixed prime p as coefficients in the matrices.

One can also make the same construction over the ring of integers $O_{\mathbb{F}}$ of a totally real number field $\mathbb{F}$ strictly containing $\mathbb{Q}$, at which point the role of p is played by $\mathfrak{p}$ a prime ideal of $O_{\mathbb{F}}$, so that the inverted elements (those allowed in denominators) are now all of $\mathfrak{p} \cap \mathbb{F}^{\times}$.

2.2 Golden gates

Golden gates are special unit groups in quaternion algebras over totally real number fields derived from the p-arithmetic groups [6]. They give variants of optimal generators for PU(2) and connect the arithmetic of quaternion algebras to quantum computation on a single qubit. The "golden" characterization is to be understood by way of an interesting link between p-arithmetic unit groups coming from unit quaternion groups and the Ramanujan graphs $X^{p,q}$, which we explicate below.

Recall once more that in classical computation, one decomposes any function into basic logical gates such as XOR, AND, and NOR, and that in quantum computation, the classical bits are replaced by qubits, which are vectors in projective Hilbert space $\mathbb{C}P^n$, and that the logical gates are *all* the elements of the projective unitary group PU(2). Let S be a subgroup of PU(2) and denote by $S^{(\ell)}$ the set of ℓ-fold products of elements in S. If $\langle S\rangle = \bigcup_{\ell \geq 0} S^{(\ell)}$ is dense in PU(2) (with respect to the standard bi-invariant metric $d^2(A, B) := 1 - \frac{|\operatorname{Tr}(A^*B)|}{2}$) then S is universal. That is, any gate can be approximated with arbitrary precision as a product of elements of S.

The notion of of a golden gate set is much stronger, requiring [18]:

1. *Optimal covering of* PU(2) *by* $\langle S\rangle$: for every ℓ the set $S^{(\ell)}$ distributes in PU(2) as a perfect sphere packing (or randomly placed points) would, up to a negligible factor.

2. *Approximation*: given $A \in \mathrm{PU}(2)$ and $\varepsilon > 0$, there is an efficient algorithm to find some $A' \in B_\varepsilon(A)$ (the ε-ball around A) such that $A' \in S^{(\ell)}$ with ℓ (almost) minimal.

3. *Compiling*: given $A \in \langle S\rangle$ as a matrix, there is an efficient algorithm to write A as a word in S of the smallest possible length.

These requirements ensure that any gate can be approximated and compiled as an efficient circuit using the gates in S.

2.3 Super golden gates

Each super golden gate set is composed of a finite group C and an involution τ, which lie in a $\mathfrak{p}$-arithmetic group for $\mathfrak{p}$ a prime ideal of the integers $O_{\mathbb{F}}$ of a totally definite quaternion algebra $\mathcal{A}$, over a totally real number field $\mathbb{F}$. We require that:

1. C acts simply transitively on the neighbors of any given vertex in X^p, the $(p+1)$-regular tree, for p the norm of $\mathfrak{p}$.

2. τ is an involution which takes a vertex to one of its neighbors.

$\mathfrak{p}$-arithmetic unit quaternion group act transitively on the vertices of the corresponding X^p. The $\mathfrak{p}$-arithmetic groups which act transitively on the vertices of X^p are called the *golden* *gates* and the $\mathfrak{p}$-arithmetic groups which act transitively on both the vertices and edges of X^p are called the *super golden gates*.

3 Icosahedral Super Golden Gates

Recall that there are only finitely many such super golden gate sets and in each case the finite group C is identified with the group of rotational symmetries of a platonic solid: for the tetrahedron, A_4; for the cube and the octahedron, S_4; and for the dodecahedron and icosahedron, A_5. Each of these finite groups can be precisely identified with a $\mathfrak{p}$-arithmetic unit quaternion group coming from one of the following quaternion algebras [6]:

- tetrahedral gates: $\mathcal{A} = \left(\frac{-1,-1}{\mathbb{Q}}\right)$;
- octahedral gates: $\mathcal{A} = \left(\frac{-1,-1}{\mathbb{Q}(\sqrt{2})}\right)$; and
- icosahedral gates: $\mathcal{A} = \left(\frac{-1,-1}{\mathbb{Q}(\sqrt{5})}\right)$.

We consider the quaternion algebra $\mathcal{A}$ over the *golden field* $\mathbb{Q}(\sqrt{5})$ [7, p.895]. A maximal order $\mathcal{O}$ in $\mathcal{A} := \left(\frac{-1,-1}{\mathbb{Q}(\sqrt{5})}\right)$ is given by the ring of icosians. The unit group $\mathcal{O}^1$ is the platonic icosahedral group, generated by $\left\langle \frac{1+i+j+k}{2}, \frac{1+\varphi^{-1}j+\varphi k}{2} \right\rangle$, where $\varphi := \frac{1+\sqrt{5}}{2}$ is the golden ratio, and $\mathcal{O}^1 \cong A_5$ In PU(2), this corresponds to

$$C_{60} = \left\langle \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}, \begin{pmatrix} 1 & \varphi - {}^{i}/_{\varphi} \\ \varphi + {}^{i}/_{\varphi} & -1 \end{pmatrix} \right\rangle =: \langle \rho, \sigma \rangle.$$

We take as our involution

$$\tau = \tau_{60} := \begin{pmatrix} 2+\varphi & 1-i \\ 1+i & -2-\varphi \end{pmatrix}.$$

For the prime ideal $\mathfrak{p} = (7+5\varphi)$ one has $\mathbb{F}_{\mathfrak{p}} \cong \mathbb{Q}_{59}$, and the generated group $\Gamma = \langle C_{60}, \tau_{60} \rangle$ is the full $(7+5\varphi)$-arithmetic group of $\mathcal{O}$. As such, we establish:

Theorem 1. *Subject to standard number-theoretic heuristic conjectures, there exists a factorization of any $g \in \mathrm{PU}(2)$ to precision ε using τ-count at most $({}^{7}/_{3} + o(1)) \log_{59}({}^{1}/_{\varepsilon^3})$.*

In particular, if g is near to a diagonal matrix then we just apply the approach in §5 (for additive error) to get a path of length at most $(1+o(1)) \log_{59}({}^{1}/_{\varepsilon^3})$, and otherwise run the approach in §6.

Notice that no other choice of golden gates is both super and has a greater logarithm base, so that these are the "best" generators of PU(2) given the present state of knowledge, as an interesting application of number theory in that geometric setting. However, we make no claim in this work as to the relevance of icosahedral golden gates to fault-tolerant quantum computing, as such relationships are not yet established; given the present stage of research into quantum computers, we offer the results of this paper merely to advance understanding of a gate set with specific known advantages, and many more unknown qualities.

4 Nearby Elements in PU(2), and Factoring in Γ

In this section, we establish a key technical lemma regarding approximations in the matrix group, and the method for exact synthesis for elements of the relevant subgroup.

For $\alpha, \beta \in \mathbb{C}$ satisfying $|\alpha|^2 + |\beta|^2 = 1$ we write

$$u(\alpha, \beta) := \begin{pmatrix} \alpha & \beta \\ -\bar{\beta} & \bar{\alpha} \end{pmatrix},$$

and for $\theta \in \mathbb{R}$ we write

$$u(\theta) := u(e^{i\theta}, 0) = \begin{pmatrix} e^{i\theta} & \\ & e^{-i\theta} \end{pmatrix}.$$

We restate below [8, Lemma 5] which holds independent of our choice of universal gate set.

Lemma 1. *Select absolute constants $\delta, \varepsilon_0 > 0$ and put $C = \sqrt{\frac{1}{2} + \frac{1}{2}\left(\frac{2+\delta}{\varepsilon_0}\right)^2}$. Take γ_1 and γ_2 in either* SU(2) *or* PU(2) *and write them as $\gamma_1 = u(\alpha_1, \beta_1)$ and $\gamma_2 = u(\alpha_2, \beta_2)$. If $||\alpha_1| - |\alpha_2|| < \varepsilon$ for some $\varepsilon < \delta$ and $\min\{|\alpha_1|, |\alpha_2|\} < \sqrt{1 - \varepsilon_0^2}$ then for*

$$\theta_1 = \frac{1}{2}(\arg\alpha_1 - \arg\alpha_2 + \arg\beta_1 - \arg\beta_2), \qquad \delta_1 = u(\theta_1)$$

$$\theta_2 = \frac{1}{2}(\arg\alpha_1 - \arg\alpha_2 - \arg\beta_1 + \arg\beta_2), \qquad \delta_2 = u(\theta_2)$$

we have the approximation $\delta_1\gamma_2\delta_2$ to γ_1, satisfying

$$d(\gamma_1, \delta_1\gamma_2\delta_2) < C\varepsilon.$$

Loosely, this result can be thought of as a sufficient condition to transform one 2 × 2 unitary into (approximately) another based only on a weak condition and by "tuning" with diagonal (rotation) matrices on either side. The condition is merely that the "starting" and "target" matrices have top-left entries nearby in absolute value, and that neither is very near to 1.

We now move to factorizing. Put $\eta = 7 + 5\varphi$ for the sequel. Observe that is is a positive real number, and the generator of $\mathfrak{p}$ above. For our purposes, we will encounter elements of $\Gamma = \langle\rho, \sigma, \tau\rangle \leq \mathrm{PU}(2)$ only as $\mathbb{Z}[\varphi]$-quaternions with norm a power of η, envisioned as elements of the Cayley graph for $\langle\rho, \sigma, \tau\rangle \leq \mathbb{Q}(\varphi)\,\mathrm{U}_2(\mathbb{Z}[\varphi])$, which [7] shows acts transitively with respect to the distance measure of τ-count, which can be detected by counting the power of η in the quaternion norm, after quotienting out $\mathbb{Z}[\varphi]$-scalars. This leads to the following factoring algorithm.

Algorithm 1. *Let C be a set of representatives of $\langle\rho, \sigma\rangle \leq \mathrm{PU}(2)$ lifted to $\mathbb{Q}(\varphi)\,\mathrm{U}_2(\mathbb{Z}[\varphi])$. Given γ a lift of some element of Γ with τ-count k, it can be factored by determining which (unique) $c \in C$ gives rise to $\gamma c\tau$ of τ-count $k-1$, which in turn requires no more than 60 multiplications of three matrices. c is found if and only if $\gamma c\tau$ has all coefficients divisble by η in $\mathbb{Z}[\varphi]$.*

This algorithm has the base case of simply comparing γ against all 60 elements of C, another trivial computational task. The idea of the general case is that γ has the representation

$$c_0 \prod_{i\in[n]} \tau c_i$$

where c_i is not the identity for $i \neq 0, n$. Then c will be projectively equal to c_n^{-1}, giving rise to

$$\gamma c\tau = c_0 \prod_{i\in[n-1]} \tau c_i \tau c_n c\tau = zpc_0 \prod_{i\in[n-1]} \tau c_i$$

for $z = \det(c_n c)$ (coprime to η).

5 Algorithm for Short Paths to Diagonal Elements

5.1 Algorithm

We proceed by, for given diagonal element $\delta = u(\theta)$ and ε, seeking $\gamma \in \Gamma$ with $d(\delta, \gamma) < \varepsilon$. Knowing that γ is projectively equal to an element

$$\begin{pmatrix} x_0 + x_1 i & x_2 + x_3 i \\ -x_2 + x_3 i & x_0 - x_1 i \end{pmatrix}$$

for $x_0, x_1, x_2, x_3 \in \mathbb{Z}[\varphi]$ satisfying

$$x_0^2 + x_1^2 + x_2^2 + x_3^2 = \eta^m \tag{4}$$

for some $m \in \mathbb{N}$, we also have that it is sufficient to satisfy

$$x_0 \cos\theta + x_1 \sin\theta \geq \eta^{m/2}(1 - 2\varepsilon^2) \tag{5}$$

(note that this is precisely [10, (13) and Problem 7.4] (see there for the derivation)), following the readily computable observation that $\|\delta - \gamma\| \geq \sqrt{2} d(\delta, \gamma)$ and setting the goal $\|\delta - \gamma\| \leq \sqrt{2}\varepsilon$, since [10] operates with respect to the operator norm. Observe, unfortunately, that (4) is a quadratic constraint and (5) is a linear constraint. We now explain how to transform them into a practicable sequence of integer programs.[a]

Fix m. We seek $x_\ell \in \mathbb{Z}[\varphi]$ satisfying (4) and (5). Define $y_\ell = {}^{x_\ell}/_{\eta^{m/2}}$. Artificially add the constraint

$$1 - \varepsilon^2 - y_1 \sin\theta \geq 0. \tag{6}$$

Observe from (4) that $y_0^2 \leq 1 - y_1^2$. Then, we have the sequence of implications (mainly by algebraic manipulation)

$$\begin{aligned} y_0 \cos\theta &\geq 1 - \varepsilon^2 - y_1 \sin\theta \geq 0 \\ y_0^2 \cos^2\theta &\geq (1 - \varepsilon^2)^2 + y_1^2 \sin^2\theta - 2(1 - \varepsilon^2) y_1 \sin\theta \\ \cos^2\theta &\geq (y_1 - (1 - \varepsilon^2)\sin\theta)^2 - (1 - \varepsilon^2)^2 \sin^2\theta + (1 - \varepsilon^2)^2 \\ \cos^2\theta(2\varepsilon^2 - \varepsilon^4) &\geq (y_1 - (1 - \varepsilon^2)\sin\theta)^2 \\ \left|y_1 - (1 - \varepsilon^2)\sin\theta\right| &\leq |\cos\theta|\sqrt{2 - \varepsilon^2}\,\varepsilon \end{aligned} \tag{7}$$

and so we have reduced our consideration to just one of the four variables. As $[\mathbb{Q}(\varphi) : \mathbb{Q}] = 2$, we explicitly work with the Galois group elements

$$\begin{aligned} \sigma_+ : 1 &\longmapsto 1 \\ \varphi &\longmapsto \varphi, \\ \sigma_- : 1 &\longmapsto 1 \\ \varphi &\longmapsto \varphi^\bullet, \end{aligned}$$

[a] Fundamentally, this is the same idea as in [10], as their study of upright ellipses accomplishes the same task as Lenstra's algorithm.

where $\varphi^\bullet$ is φ's Galois conjugate $\frac{1-\sqrt{5}}{2} = 1 - \varphi$, both of which are real embeddings, yielding the additional constraints

$$|\sigma_\pm y_1| \leq 1. \tag{8}$$

Now, putting $x_1 = c + d\varphi$ for some $c, d \in \mathbb{Z}$, (6), (7), and (8) transform into

Problem 1. *Given m, find $(c, d) \in \mathbb{Z}^2$ satisfying the five linear constraints*

$$\begin{aligned} (c + d\varphi)\sin\theta &\leq \eta^{m/2}(1 - \varepsilon^2), \\ |c + d\sigma_\pm\varphi| &\leq (\sigma_\pm\eta)^{m/2}, \\ \left|c + d\varphi - \eta^{m/2}(1 - \varepsilon^2)\sin\theta\right| &\leq \eta^{m/2}|\cos\theta|\sqrt{2 - \varepsilon^2}\varepsilon. \end{aligned}$$

The existence of such a pair is determinable in $O(\operatorname{poly} m)$ time, by Lenstra's algorithm [12, 13].

Putting $x_0 = a + b\varphi$ for some $a, b \in \mathbb{Z}$, for a particular solution to Problem 1, (4) and (5) transform into

Problem 0. *Given m and $x_1 = c + d\varphi$, find $(a, b) \in \mathbb{Z}^2$ satisfying the four linear constraints*

$$\begin{aligned} |a + b\sigma_\pm\varphi| &\leq (\sigma_\pm\eta)^{m/2}\sqrt{1 - (\sigma_\pm x_1)^2}, \\ (a + b\varphi)\cos\theta &\leq \eta^{m/2}(1 - x_1\sin\theta), \\ (a + b\varphi)\cos\theta &\geq \eta^{m/2}(1 - \varepsilon^2 - x_1\sin\theta). \end{aligned}$$

Again, the existence of such a pair is determinable in $O(\operatorname{poly} m)$ time, by Lenstra's algorithm.

Finally, if we have solutions to Problem 0 and Problem 1, we seek to solve

Problem 23. *Given m, $x_0 = a + b\varphi$, and $x_1 = c + d\varphi$, find $(x_2, x_3) \in \mathbb{Z}[\varphi]^2$ satisfying* (4).

Here we change techniques. The objective is to write $\eta^m - x_0^2 - x_1^2$ as a sum of squares in $\mathbb{Z}[\varphi]$. Assuming efficient factorization in $\mathbb{Z}$ (or $\mathbb{Z}[\varphi]$, also a PID), Problem 23 is efficiently solvable via Theorem 3 (the general approach being basically identical to the classical algorithms for $\mathbb{Z}[i]$ or $\mathbb{Z}[e^{i\pi/8}]$, cf. [10, §C] for the latter).

We have succeeded in the core of the algorithm. To handle given δ, begin by fixing $m = 0$. For given m, attempt to solve Problem 1; for each solution, attempt to solve Problem 0; for each solution, attempt to solve Problem 23. If this results in a tuple (x_0, x_1, x_2, x_3) satisfying all three problems, halt and return

$$\frac{1}{\eta^{m/2}}\begin{pmatrix} x_0 + x_1 i & x_2 + x_3 i \\ -x_2 + x_3 i & x_0 - x_1 \end{pmatrix}.$$

Otherwise, if all possibilities have been exhausted, increment m.

5.2 Analysis

We refer the reader to [7, §2.3]'s analysis of timing and correctness for Ross and Selinger's algorithm generalized to any golden gate set, where the conclusion is that for desired precision ε, the factorization length becomes $(1 + o(1))\log_{59}(1/\varepsilon^3)$ with required computational time remaining $O(\operatorname{poly}\log(1/\varepsilon))$.

6 Algorithm For Short Paths

Here we largely adopt the structure and wording from [8, §4 and §5], as the core concepts and heuristics that make the algorithm work are the same in the icosahedral setting.

6.1 Algorithm

Select absolute constants $\tilde{\varepsilon}, \varepsilon_0 > 0$. Take any $g = u(\alpha, \beta) \in \mathrm{PU}(2)$ where $|\alpha| < \sqrt{1-\varepsilon_0^2}$, and pick $\varepsilon < \tilde{\varepsilon}$. We wish to approximate g in d using $\gamma \in \Gamma$ of the form

$$\gamma = \frac{1}{\eta^{k/2}} \begin{pmatrix} x_0 + x_1 i & x_2 + x_3 i \\ -x_2 + x_3 i & x_0 - x_1 i \end{pmatrix} \tag{9}$$

having k, the factorization length, minimized, and so we begin with $k = 0$. (We also have $x_0, x_1, x_2, x_3 \in \mathbb{Z}[\varphi]$.) In particular, the objective is to approximate g as $\gamma_1 \gamma \gamma_2$ where $\gamma_1, \gamma_2 \in \Gamma$ approximate well-chosen diagonals computable using §5, and $\gamma \in \Gamma$ has factorization computable using Algorithm 1. We will see that γ is designed to have factorization typically shorter than that of γ_1 and γ_2, giving rise to the desired improvement.

In order to apply Lemma 1 we need to have $\left|\frac{x_0+x_1 i}{\eta^{k/2}}\right| = \sqrt{\frac{x_0^2+x_1^2}{\eta^k}}$ near $|\alpha|$ (that is, within ε). Because $\left|\frac{x_0+x_1 i}{\eta^{k/2}}\right| + |\alpha| \geq |\alpha|$ which is fixed, it suffices to find candidate values for $x_0, x_1 \in \mathbb{Z}[\varphi]$ with $\left|\left|\frac{x_0+x_1 i}{\eta^{k/2}}\right|^2 - |\alpha|^2\right| < \varepsilon|\alpha|$, rewritten to

$$\left|x_0^2 + x_1^2 - |\alpha|^2 \eta^k\right| < \varepsilon |\alpha| \eta^k. \tag{10}$$

Viewing γ as an element of SU(2), we also have $\det \gamma = 1$, i.e. $x_0^2 + x_1^2 + x_2^2 + x_3^2 = \eta^k$. As $x_\ell \in \mathbb{Z}[\varphi] \subset \mathbb{Q}(\varphi) \subset \mathbb{R}$, it follows that $\sigma_\pm(x_0^2 + x_1^2) + \sigma_\pm(x_2^2 + x_3^2) = \sigma_\pm(\eta^k)$, and so

$$\sigma_\pm(x_0^2 + x_1^2) \leq (\sigma_\pm \eta)^k. \tag{11}$$

Now, let $m = x_0^2 + x_1^2 \in \mathbb{Z}[\varphi]$. Considering $\mathbb{Z}[\varphi]$ as an integer lattice, we adapt (10) and (11) and seek to solve

$$\left|m - |\alpha|^2 \eta^k\right| < \varepsilon |\alpha| \eta^k \tag{12}$$

$$m \leq \eta^k \tag{13}$$

$$m^\bullet \leq (\eta^\bullet)^k \tag{14}$$

$$m, m^\bullet \geq 0 \tag{15}$$

which are convex constraints on m's lattice components. Since this is an integer programming problem in two dimensions, we apply Lenstra's algorithm [12, 13] to efficiently list all such lattice points m. For each m, using Theorem 3, we attempt to write m as a sum of two squares; if possible, say $m = x_0^2 + x_1^2$, we then attempt to write $\tilde{m} = \eta^k - m$ as a sum of two squares. If possible, say $\tilde{m} = x_2^2 + x_3^2$, so we simply halt and return γ corresponding to (9). However, if $\tilde{m}$ may not be represented as a sum of two squares, we simply move on to the next value of m and try this process again. If this fails for all m arising from k, we increment k and run Lenstra's algorithm for the new inequalities.

Supposing we have halted and constructed γ, we compute δ_1 and δ_2 guaranteed by Lemma 1. These are efficiently approximable by §5 to γ_1 and γ_2, respectively. Chaining together the three approximations as $\gamma_1 \gamma \gamma_2$ gives the final desired approximation to g.

6.2 Analysis

We begin the analysis by establishing the τ-count and tightness of the approximation. In particular, $d(\gamma_1, \delta_1) < \varepsilon$ and $d(\gamma_2, \delta_2) < \varepsilon$ with factorization lengths of γ_ℓ each up to $(1+o(1)) \log_{59}(1/\varepsilon^3)$. By Lemma 1, $d(\gamma, \delta_1\gamma\delta_2) < C\varepsilon$. Therefore, $d(g, \gamma_1\gamma\gamma_2) < (C+2)\varepsilon$ (by the triangle inequality) and since γ has a factorization of τ-count up to $\left(\frac{1}{3} + o(1)\right) \log_{59}(1/\varepsilon^3)$, this constitutes a factorization of an element in g's neighborhood of τ-count up to $\left(\frac{7}{3} + o(1)\right) \log_{59}(1/\varepsilon^3)$.

The efficiency of this algorithm—that is, that it runs in time $O(\operatorname{poly}\log(1/\varepsilon))$—is because we expect to halt when $59^k\varepsilon \in O(1)$ (so only $k \approx \frac{1}{3}\log_{59}(1/\varepsilon^3)$ calls are expected), and only call polynomially-many polynomial-time subroutines. The dominant subroutines are calls to Lenstra's algorithm which as shown in [12] runs in time polynomial in the size of the constraints for any fixed dimension n. Indeed, here we have only $n = 2$ dimensions, $m = 6$ linear constraints (two per absolute value), and the largest value a in the constraints is η^k, so the runtime is polynomial in $nm \log a \in \Theta(k)$.

The reason we expect to halt when $59^k\varepsilon \in O(1)$ is that heuristically, we expect to halt when the area enclosed by (12)–(15) is $O(\operatorname{poly}\log(1/\varepsilon))$. Conveniently, the region is a rectangle since the vectors $\langle 1, \varphi\rangle$ and $\langle 1, \varphi^\bullet\rangle$ are orthogonal, so assuming in the limit that $\varepsilon \ll \min\left\{|\alpha|, \frac{1-|\alpha|^2}{|\alpha|}\right\}$ (so that (13) and the "bottom" inequality of (12) are redundant), we compute length-times-width of

$$\frac{2\varepsilon|\alpha|\eta^k}{\sqrt{1+\varphi^2}} \cdot \frac{(\eta^\bullet)^k}{\sqrt{1+(1-\varphi)^2}} = \frac{2|\alpha|}{\sqrt{5}} \cdot 59^k\varepsilon \in O(\operatorname{poly}\log(1/\varepsilon))$$

whence we find $k \in (1 + o(1)) \log_{59}(1/\varepsilon)$ as expected.

When attempting to write elements of $\mathcal{O}$ as a sum of two squares, we primarily rest on a belief, in the style of Cramér's conjecture and a conjecture of Sardari [26, (∗)] that sums of squares are dense in $\mathbb{N}$. Seeking to analogize [26, (∗)] in particular, we note that the operative aspect is that a dense cluster of lattice points will represent at least one sum of two squares, and that such a point thus will be found quickly through Lenstra's algorithm.

The significance of this result is to accomplish a factorization in PU(2) in analogue to [8], but using a gate set with additional desirable properties beyond those enjoyed by the Clifford+T gates.

7 Implementation Details and Examples

Our algorithm has been implemented for proof-of-concept purposes in Python, and the code is available at `https://math.berkeley.edu/ zstier/icosahedral`. Included in this implementation are:

- An implementation of Lenstra's algorithm for special cases (`convex.py`).
- The rings $\mathbb{Z}[\varphi]$, $\mathbb{Z}[i, \varphi]$, and $H(\mathbb{Z}[\varphi])$ (`rings.py` and `quaternions.py`).
- Solutions to sum-of-two-square problems in $\mathbb{Z}[\varphi]$ (`rings.py`).
- Factorization of elements of $H(\mathbb{Z}[\varphi])$ which are of norm a power of η (`quaternions.py`).
- Efficient factorization of diagonal elements of PU(2), as outlined in §5 (`diagonal.py`).

- Efficient factorization of general elements of PU(2), as outlined in §6 (`approx.py`).

In the remainder of this section, we demonstrate the efficacy of this factorization technique on the two more “classical” single-qubit quantum gate generators; recall

$$H = \frac{i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } T = \begin{pmatrix} e^{i\pi/8} & \\ & e^{-i\pi/8} \end{pmatrix}.$$

In both cases, pick precision $\varepsilon = 1/10^{10}$.

For the first example of T, we yield

$$\begin{aligned} T &\approx \xi_0\tau\xi_1\tau\xi_2\tau\xi_3\tau\xi_4\tau\xi_5\tau\xi_6\tau\xi_7\tau\xi_8\tau\xi_9\tau\xi_{10}\tau\xi_{11}\tau\xi_{12}\tau\xi_{13}\tau\xi_{14}\tau\xi_{15}\tau\xi_{16}\tau\xi_{17}\tau\xi_{18}\tau\xi_{19} \\ &= (\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho \\ &\quad \sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho) \\ &\quad \tau(\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho) \\ &\quad \tau(\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho \\ &\quad \sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho\sigma) \end{aligned}$$

where $\xi_i \in \langle\sigma, \rho\rangle$ correspond to the parenthesized terms following; this achieves precision in d of $1.28/10^{10}$. The τ-count is 19; compare this with the predicted $\log_{59}(1/\varepsilon^3) \approx 16.9$. We remark that the discrepancy can be attributed to computational limitations: at key steps in the algorithm, we must run the Tonelli–Shanks algorithm for finding quadratic residues modulo some prime q, which has worst-case behavior $\Theta(q)$. Therefore it is only practical to abandon on instances where some prime factor is at least 10^6, something that occurred more than 328 times before the above-stated approximation was found.

For the second example of H, the central element is determined to be the following:

$$\begin{aligned} \gamma &= \xi_0\tau\xi_1\tau\xi_2\tau\xi_3\tau\xi_4\tau\xi_5\tau\xi_6\tau\xi_7\tau\xi_8\tau\xi_9 \\ &= (\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma)\tau \\ &\quad (\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho \\ &\quad \sigma\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\rho) \end{aligned}$$

(ξ_i serves the same function as above). It has τ-count 9; compare this with the predicted $\frac{1}{3}\log_{59}(1/\varepsilon^3) \approx 5.6$. As before, the discrepancy can be attributed to non-vanishing of $o(1)$ factors for explicit ε and to having to abandon possibilities with very large prime factors.

The outer diagonal elements are determined to be the following:

$$\begin{aligned}
\gamma_1 &\approx \tilde{\xi}_0\tau\tilde{\xi}_1\tau\tilde{\xi}_2\tau\tilde{\xi}_3\tau\tilde{\xi}_4\tau\tilde{\xi}_5\tau\tilde{\xi}_6\tau\tilde{\xi}_7\tau\tilde{\xi}_8\tau\tilde{\xi}_9\tau\tilde{\xi}_{10}\tau\tilde{\xi}_{11}\tau\tilde{\xi}_{12}\tau\tilde{\xi}_{13}\tau\tilde{\xi}_{14}\tau\tilde{\xi}_{15}\tau\tilde{\xi}_{16}\tau\tilde{\xi}_{17}\tau\tilde{\xi}_{18}\\
&= (\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\sigma\rho\sigma\sigma)\tau(\rho\sigma\sigma\rho\sigma\\
&\quad \rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\rho\sigma)\tau(\sigma\rho\sigma\rho)\tau(\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma)\\
&\quad \tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\\
&\quad \rho\sigma\sigma\rho)\tau(\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\sigma)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\\
&\quad \tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma)\\
\gamma_2 &\approx \hat{\xi}_0\tau\hat{\xi}_1\tau\hat{\xi}_2\tau\hat{\xi}_3\tau\hat{\xi}_4\tau\hat{\xi}_5\tau\hat{\xi}_6\tau\hat{\xi}_7\tau\hat{\xi}_8\tau\hat{\xi}_9\tau\hat{\xi}_{10}\tau\hat{\xi}_{11}\tau\hat{\xi}_{12}\tau\hat{\xi}_{13}\tau\hat{\xi}_{14}\tau\hat{\xi}_{15}\tau\hat{\xi}_{16}\tau\hat{\xi}_{17}\tau\hat{\xi}_{18}\\
&= (\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\sigma\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\\
&\quad \sigma\rho\sigma\rho)\tau(\sigma\sigma\rho\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\sigma)\tau(\sigma\rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\\
&\quad \rho\sigma\sigma\rho\sigma\rho)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\sigma)\tau(\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\rho\sigma\sigma\rho\\
&\quad \sigma\rho)\tau(\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma\rho\sigma\sigma\rho)\tau(\rho\sigma\sigma\rho\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\\
&\quad \sigma\rho\sigma\sigma)\tau(\rho\sigma\rho\sigma\sigma\rho\sigma\rho\sigma\sigma)\tau(\sigma\sigma\rho\sigma\rho\sigma\sigma\rho\sigma)\tau(\sigma\rho)\tau(\rho)
\end{aligned}$$

($\tilde{\xi}_i$, $\hat{\xi}_i$ serves the same function as above). They both have τ-counts of 18, with 75 collective abandoned cases; compare this with the predicted $\log_{59}({}^1\!/\!{}_{\varepsilon^3}) \approx 16.9$. Multiplying out $\gamma_1\gamma\gamma_2$ gives distance in d of ${}^{1.28}\!/\!{}_{10^{10}}$ (that this difference equals the previous one is a coincidence; they disagree at the third decimal place).

8 Upper-Bounding Fault-Tolerant Resources

We are grateful to one of the anonymous referees from *Quantum Information and Computation* for the remark which comprises this section, reproduced here for the reader's benefit with only light editing.

Note that it is possible to use existing results to implement super golden gates on common fault-tolerant quantum computers, as well as any gate specified by a matrix with entries in some number field. Therefore, it is theoretically possible to upper-bound the cost of icosahedral golden gates.

Consider a gate U with entries in some number field K. This gate can be synthesized in three steps using three well-known results.

First, reduce it to implementing an n-qubit unitary U' with entries in one of the fields $\mathbb{Q}(\zeta_8)$, $\mathbb{Q}(i)$, or Q, by employing the Catalytic Embedding idea presented in [27, 28].

Second, synthesize U' using an $(n+2)$-qubit unitary U'' with entries in the rings $\mathbb{Z}[\zeta_8, {}^1\!/\!{}_2]$, $\mathbb{Z}[i, {}^1\!/\!{}_2]$, or $\mathbb{Z}[{}^1\!/\!{}_2]$, utilizing [29, Theorem 18].

Third, employ exact synthesis techniques for unitaries [30, 31] with entries in $\mathbb{Z}[\zeta_8, {}^1\!/\!{}_2]$, $\mathbb{Z}[i, {}^1\!/\!{}_2]$, or $\mathbb{Z}[{}^1\!/\!{}_2]$ to find a Clifford and T circuit for the unitary U''. Additionally, synthesize the catalyst state up to the required accuracy using existing approximate synthesis techniques. The cost of the catalyst state can be neglected since it is reused.

Furthermore, the results presented in [32] can be utilized to numerically establish lower bounds on the T-cost of synthesizing U''. This approach allows us to determine if icosahedral golden gates can become practical if multi-qubit synthesis algorithms are improved.

9 Concluding Remarks

This work represents a theoretical, heuristic, and proof-of-concept demonstration of a state-of-the-art methodology to construct single-qubit quantum gates, optimizing for using as few expensive gates as possible, and in particular representing an improvement of $\log_2 59 \approx 5.9$ over the previous best method [8]. However, the area of efficient quantum hardware selection is far from fully explored. While [7] demonstrates that efficiently computing a length-$\log_p(1/\varepsilon^3)$ factorization is NP-complete (for an arbitrary PU(2)-element into golden gates associated to prime p), it may still be possible to achieve length-$c\log_p(1/\varepsilon^3)$ factorizations for $c \in (1, 7/3)$. Another possibility is in the study of multiple qubits simultaneously, as has been initiated for PU(3) by Evra and Parzanchevski [33]. Further, the relationship of the icosahedral gates to fault-tolerant quantum computing is as of yet not well understood.

Acknowledgements

TRB was supported, in part, by the Institute for Advanced Study and Medgar Evers College. ZS was supported by a National Science Foundation graduate research fellowship, grant numbers DGE-1752814/2146752. We are deeply indebted to Peter Sarnak for introducing us to this problem and for the many conversations that led to the production of this paper. We are also grateful to Carlos Esparza, Subhayan Roy Moulik, John Voight, and the two anonymous referees from *Quantum Information and Computation* for their suggestions.

References

1. A. Lubotzky, R. Phillips, and P. Sarnak (1986), *Explicit expanders and the Ramanujan conjectures*, Proceedings of the eighteenth annual ACM Symposium on Theory of Computing, pages 240–246.
2. A. Lubotzky, R. Phillips, and P. Sarnak (1986), *Hecke operators and distributing points on the sphere. I*, Communications on Pure and Applied Mathematics, 39(S1):S149–S186.
3. A. Lubotzky, R. Phillips, and P. Sarnak (1987), *Hecke operators and distributing points on* S^2*. II*, Communications on Pure and Applied Mathematics, 40(4):401–420.
4. A. Lubotzky, R. Phillips, and P. Sarnak (1988), *Ramanujan graphs*, Combinatorica, 8(3):261–277.
5. M. A. Nielsen and I. L. Chuang (2010), *Quantum Computation and Quantum Information: 10th Anniversary Edition.*
6. P. Sarnak (2015), *Letter to Aaronson and Pollington on the Solovay–Kitaev Theorem and Golden Gates.*
7. O. Parzanchevski and P. Sarnak (2018), *Super Golden Gates for PU(2)*, Advances in Mathematics, 327:869–901.
8. Z. Stier (2021), *Short Paths in* PU(2), Quantum Information and Computation, 21(9&10).
9. P. Niemann, R. Wille, and R. Drechsler (2020), *Advanced exact synthesis of Clifford+T circuits*, Quantum Information Processing, 19(9):1–23.
10. N. J. Ross and P. Selinger (2016), *Optimal ancilla-free Clifford+T approximation of z-rotations*, Quantum Information and Computation, 16(11&12).
11. V. Kliuchnikov, D. Maslov, and M. Mosca (2012), *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*, Quantum Information and Computation, 13(7&8).
12. H. W. Lenstra, Jr. (1983), *Integer Programming with a Fixed Number of Variables*, Mathematics of Operations Research, 8(4).
13. A. Paz (1983), *A simplfied version of H.W. Lenstra's integer programming algorithm and some applications.*
14. E. Carvalho Pinto and C. Petit (2018), *Better path-finding algorithms in LPS Ramanujan graphs*, Journal of Mathematical Cryptology, 12(4).

15. M.-F. Vignéras (1980), *Arithmétique des algèbres de quaternions*, Lecture Notes in Mathematics, 800.
16. J. Voight (2021), *Quaternion algebras.*
17. D. W. Morris (2001), *Introduction to arithmetic groups*, arXiv:math/0106063.
18. A. Lubotzky and O. Parzanchevski (2020), *From Ramanujan graphs to Ramanujan complexes*, Philosphical Transactions of the Royal Society A, 378(2163):20180445.
19. T. Miyake (2006), *Modular forms.*
20. S. Johansson (1997), *A description of quaternion algebras.*
21. R. Dubisch (1947), *The Wedderburn structure theorems*, The American Mathematical Monthly, 54(5):253–259.
22. D. W. Lewis (2006), *Quaternion algebras and the algebraic legacy of Hamiltonian's quaternions*, Irish Mathematical Society Bulletin, 57:41–64.
23. A. Strömbergsson (2000), *An application of an explicit trace formula to a well-known spectral correspondence on quaternion groups.*
24. G. P. Davidoff, P. Sarnak, and A. Valette (2003), *Elementary number theory, group theory, and Ramanujan graphs.*
25. N. T. Sardari (2015), *Diameter of Ramanujan graphs and random Cayley graphs with numerics*, arXiv:1511.09340.
26. N. T. Sardari (2019), *Complexity of Strong Approximation on the Sphere*, International Mathematics Research Notices.
27. A. Glaudell (2022), *Catalytic Embeddings: Linking Gate Teleportation to Circuit Synthesis*, talk at Quantum Information Processing.
28. M. Amy, M. Crawford, A. N. Glaudell, M. L. Macasieb, S. S. Mendelson, and N. J. Ross (2023), *Catalytic Embeddings of Quantum Circuits*, arXiv:2305.07720.
29. A. Bocharov, M. Roetteler, and K. M. Svore (2015), *Efficient Synthesis of Universal Repeat-Until-Success Quantum Circuits*, Physical Review Letters, 114(8).
30. B. Giles and P. Selinger (2013), *Exact synthesis of multiqubit Clifford+T circuits*, Physical Review A, 87(3).
31. M. Amy, A. Glaudell, and N. J. Ross (2020), *Number-Theoretic Characterizations of Some Restricted Clifford+T Circuits*, Quantum, 4(252).
32. M. Beverland, E. Campbell, M. Howard, and V. Kliuchnikov (2020), *Lower bounds on the non-Clifford resources for quantum computations*, Quantum Science and Technology, 5(3).
33. S. Evra and O. Parzanchevski (2022), *Ramanujan Complexes and Golden Gates in* PU(3), Geometric and Functional Analysis.
34. The LMFDB Collaboration, number field 4.0.400.1: $\mathbb{Q}(i, \sqrt{5})$.

Appendix A

Here, we show that $\mathbb{Q}(i, \varphi)$ is norm-Euclidean.

Put $R = \mathbb{Z}[i, \varphi]$ and $K = \mathbb{Q}(i, \varphi)$.

Proposition 1. *R is the ring of integers of K.*

Proof. Consider the canonical $\mathbb{Z}$-basis of R, namely $\{1, \varphi, i, i\varphi\}$. One readily computes its discriminant to be 400, equal to K's, as per [34]. □

Consider the norm function $N = \left|N_{K/\mathbb{Q}}\right|$, defined on K and taking integral values on R. (The absolute value here is customary but superfluous, as one of K's $\mathbb{Q}$-automorphisms is complex conjugation.)

In what follows, balls $\mathcal{B}(r)$ of radius r are centered at the origin and closed and with respect to the ℓ_∞-norm.

We shall treat K interchangably with its formulation as the $\mathbb{Q}$-space $\mathbb{Q}^4$.

Theorem 2. *R is norm-Euclidean; that is, R is a Euclidean domain with respect to the Euclidean function N.*

Proof. We use the standard reformulation that R is norm-Euclidean if and only if for all $\alpha \in K$ there is $\beta \in R$ for which $N(\alpha - \beta) < 1$. Therefore we shall attack the latter statement.

Put $\alpha = w + x\varphi + yi + zi\varphi \in K$. Then, we readily compute

$$\begin{aligned} N(\alpha) = {} & w^4 + 2w^3x - w^2x^2 - 2wx^3 + x^4 + 2w^2y^2 + 2wxy^2 + 3x^2y^2 + y^4 + 2w^2yz \\ & - 8wxyz - 2x^2yz + 2y^3z + 3w^2z^2 - 2wxz^2 + 2x^2z^2 - y^2z^2 - 2yz^3 + z^4. \end{aligned}$$

Define $\|\alpha\| = \max\{|w|, |x|, |y|, |z|\}$. Then we can also compute, for fixed α and any other $\beta \in K$ with $\|\beta\| \to 0$,

$$N(\alpha + \beta) \leq N(\alpha) + O(\|\beta\|), \tag{A.1}$$

and we shall presently obtain an effective, non-asymptotic form of this fact.

Viewing $K \cong \mathbb{Q}^4$ as $\mathbb{Q}$-spaces and R as the standard lattice, we can translate any element of K using R to one with ℓ_∞-norm at most $1/2$ (that is, $\mathcal{B}(1/2)$), by subtracting off the "rounded" element—round each component to the nearest integer. So, it remains to verify whether every element $\alpha \in \mathcal{B}(1/2)$ of the vector space has $N(\alpha) < 1$ (which turns out to not actually hold!). To attempt to accomplish this, we look at a refinement of R. In particular, consider $\Lambda = \frac{1}{n}\mathbb{Z}^4$ for well-chosen n. We cover $\mathcal{B}(1/2)$ with Λ-translates of $\mathcal{B}(1/2n)$, so that for each $\alpha \in \Lambda$, for all $\beta \in \alpha + \mathcal{B}(1/2n)$, by (A.1) we have $N(\beta) \leq N(\alpha) + O(1/2n)$. The balancing act, then, is to choose n small enough so that $\#(\Lambda \cap \mathcal{B}(1/2))$ is manageable for a computer, but large enough so that $1/2n$ is sufficiently small. There is a catch, where it is not actually the case that we can ensure that $N(\alpha) + O(1/2n) < 1$ for all $\alpha \in \Lambda \cap \mathcal{B}(1/2)$; however, for those α which violate this condition, we can attempt to circumvent the obstruction by translating to $(\alpha + \delta) + \mathcal{B}(1/2n)$ (where $\delta \in \mathbb{Z}^4$) and then taking N.[b]

We now establish (A.1), after which we will be able to select n. Keeping $\alpha = w + x\varphi + yi + zi\varphi$, put $\beta := d_1 + d_2\varphi + d_3 i + d_4\varphi$. Fully written out, $N(\alpha + \beta)$ is (the absolute value of) a degree-four polynomial in eight variables with 170 total terms, so we spare the reader

[b] As it turns out, picking δ so that it shifts exactly one component towards 0 by exactly 1 is sufficient when any δ is necessary at all.

its full statement. Letting $\|\beta\| \leq \varepsilon$,[c] however, we have that

$$\begin{aligned}
N(\alpha+\beta) \leq N(\alpha) \\
&+ 2\varepsilon\Big(\big|2w^3 + 3w^2x - wx^2 - x^3 + 2wy^2 + xy^2 + 2wyz + 3wz^2 - xz^2 - 4xyz\big| \\
&\quad + \big|w^3 - w^2x - 3wx^2 + 2x^3 + wy^2 + 3xy^2 - 4wyz - 2xyz - wz^2 + 2xz^2\big| \\
&\quad + \big|2w^2y + 2wxy + 3x^2y + 2y^3 + w^2z - 4wxz - x^2z + 3y^2z - yz^2 - z^3\big| \\
&\quad + \big|w^2y - 4wxy - x^2y + y^3 + 3w^2z - 2wxz + 2x^2z - y^2z - 3yz^2 + 2z^3\big|\Big) \\
&+ \varepsilon^2\Big(\big|6w^2 + 6wx - x^2 + 2y^2 + 2yz + 3z^2\big| + 2\big|3w^2 - 2wx - 3x^2 + y^2 - 4yz - z^2\big| \\
&\quad + \big|-w^2 - 6wx + 6x^2 + 3y^2 - 2yz + 2z^2\big| + \big|2w^2 + 2wx + 3x^2 + 6y^2 + 6yz - z^2\big| \\
&\quad + 2\big|w^2 - 4wx - x^2 + 3y^2 - 2yz - 3z^2\big| + \big|3w^2 - 2wx + 2x^2 - 6yz + 6z^2 - y^2\big| \\
&\quad + 4|2wy + xy + wz - 2xz| + 4|wy + 3xy - 2wz - xz| + 4|wy - 2xy + 3wz - xz| \\
&\quad + 4|-2wy - xy - wz + 2xz|\Big) \\
&+ 2\varepsilon^3\big(|w+x| + 2|3w-x| + 2|w+3x| + 2|2x-w| + 3|2w+x| + 2|w-2x| \\
&\quad + 2|2z-y| + 2|y+3z| + 2|y-2z| + 2|3y-z| + 4|2y+z|\big) \\
&+ 40\varepsilon^4.
\end{aligned}$$

Then, picking $n := 6$ and $\varepsilon := 1/12$, for each $\alpha \in \Lambda \cap \mathcal{B}(1/2)$ as well as $\alpha - (\operatorname{sgn} w, 0, 0, 0)$, $\alpha - (0, \operatorname{sgn} x, 0, 0)$, $\alpha - (0, 0, \operatorname{sgn} y, 0)$, and $\alpha - (0, 0, 0, \operatorname{sgn} z)$, we test whether the right-hand side of the above is bounded by 1 for any of those five choices. The code appearing in Figure A.1 verifies that this indeed comes to pass. □

Remark 1. *It would appear that this method would readily adapt to a computation to determine that additional number fields are norm-Euclidean, so long as one knows a basis for its ring of integers and correspondingly computes an appropriate analogue to the local effective bound* `KQnorm(w,x,y,z,r)`*. Unfortunately we have been unable to reproduce this success with any other biquadratic number field of the form* $\mathbb{Q}(i, \sqrt{n})$*,* $n > 5$*.*

Appendix B

Here, we study irreducible elements and sums of two squares in $\mathbb{Z}[\varphi]$.

As in [10, §C], here we shall summarize results about classifying irreducible elements in $\mathbb{Z}[\varphi]$ and an efficient algorithm for writing certain elements of $\mathbb{Z}[\varphi]$ as a sum of two squares.

For this section, let $N = N_{\mathbb{Q}(\varphi)/\mathbb{Q}}$ (in contrast to $N = N_{\mathbb{Q}(i,\varphi)/\mathbb{Q}}$ as in §1). We readily compute $N(a + b\varphi) = a^2 + ab - b^2$.

Recall that $\mathbb{Z}[\varphi]$ is a Euclidean domain with respect to $|N|$, and that $\mathbb{Z}[\varphi]^\times = \langle \pm\varphi \rangle$ (cf. [7, §4.1.4]). Also recall that $\varphi^\bullet$ is φ's Galois conjugate, which happens to equal $1 - \varphi$. Extend $^\bullet$ $\mathbb{Q}$-linearly to all of $\mathbb{Q}(\varphi)$.

Proposition 2. *Let p be irreducible in $\mathbb{Z}$. If $p \equiv \pm 2 \pmod 5$ then p is irreducible in $\mathbb{Z}[\varphi]$. If $p \equiv \pm 1 \pmod 5$ then there is an algorithm (running in time $O(\operatorname{poly}\log p)$) to compute a*

[c] So that $|d_\ell| \leq \varepsilon$ for all $\ell \in [4]$.

```python
import numpy as np
def KQnorm(w,x,y,z,r):
    p0 = abs(w**4 + 2 * w**3 * x - w**2 * x**2 - 2 * w * x**3 + x**4 \
        + 2 * w**2 * y**2 + 2 * w * x * y**2 + 3 * x**2 * y**2 \
        + y**4 + 2 * w**2 * y * z - 8 * w * x * y * z \
        - 2 * x**2 * y * z + 2 * y**3 * z + 3 * w**2 * z**2 \
        - y**2 * z**2 - 2 * y * z**3 + z**4 - 2 * w * x * z**2 \
        + 2 * x**2 * z**2)
    p1 = 2 * r * (abs(2 * w**3 + 3 * w**2 * x - w * x**2 - x**3 \
        + 2 * w * y**2 + x * y**2 + 2 * w * y * z + 3 * w * z**2 \
        - x * z**2 - 4 * x * y * z) + abs(w**3 - w**2 * x \
        - 3 * w * x**2 + 2 * x**3 + w * y**2 + 3 * x * y**2 \
        - 4 * w * y * z - 2 * x * y * z - w * z**2 + 2 * x * z**2) \
        + abs(2 * w**2 * y + 2 * w * x * y + 3 * x**2 * y + 2 * y**3 \
        + w**2 * z - 4 * w * x * z - x**2 * z + 3 * y**2 * z \
        - y * z**2 - z**3) + abs(w**2 * y - 4 * w * x * y - x**2 * y \
        + y**3 + 3 * w**2 * z - 2 * w * x * z + 2 * x**2 * z \
        - y**2 * z - 3 * y * z**2 + 2 * z**3))
    p2 = r**2 * (abs(6 * w**2 + 6 * w * x - x**2 + 2 * y**2 \
        + 2 * y * z + 3 * z**2) + abs(6 * w**2 - 4 * w * x \
        - 6 * x**2 + 2 * y**2 - 8 * y * z - 2 * z**2) \
        + abs( - w**2 - 6 * w * x + 6 * x**2 + 3 * y**2 - 2 * y * z \
        + 2 * z**2) + abs(2 * w**2 + 2 * w * x + 3 * x**2 + 6 * y**2 \
        + 6 * y * z - z**2) + abs(2 * w**2 - 8 * w * x - 2 * x**2 \
        + 6 * y**2 - 4 * y * z - 6 * z**2) + abs(3 * w**2 \
        - 2 * w * x + 2 * x**2 - 6 * y * z + 6 * z**2 - y**2) \
        + abs(8 * w * y + 4 * x * y + 4 * w * z - 8 * x * z) \
        + abs(4 * w * y + 12 * x * y - 8 * w * z - 4 * x * z) \
        + abs(4 * w * y - 8 * x * y + 12 * w * z - 4 * x * z) \
        + abs( - 8 * w * y - 4 * x * y - 4 * w * z + 8 * x * z))
    p3 = 2 * r**3 * (abs(w + x) + 2 * abs(3 * w - x) \
        + 2 * abs(w + 3 * x) + 2 * abs(2 * x - w) \
        + 3 * abs(2 * w + x) + 2 * abs(w - 2 * x) \
        + 2 * abs(2 * z - y) + 2 * abs(y + 3 * z) \
        + 2 * abs(y - 2 * z) + 2 * abs(3 * y - z) \
        + 4 * abs(2 * y + z))
    p4 = 40 * r**4
    return p0 + p1 + p2 + p3 + p4
n = 7
r = 1.0/(2*(n-1))
l = np.linspace(-0.5,0.5,n)
for a in l:
    for b in l:
        for c in l:
            for d in l:
                if KQnorm(a,b,c,d,r) >= 1 \
                    and KQnorm(a-np.sign(a),b,c,d,r) >= 1 \
                    and KQnorm(a,b-np.sign(b),c,d,r) >= 1 \
                    and KQnorm(a,b,c-np.sign(c),d,r) >= 1 \
                    and KQnorm(a,b,c,d-np.sign(d),r) >= 1:
                    print (a,b,c,d)
```

Fig. A.1. Code used to prove Theorem 2. It prints 4-tuples corresponding to points whose norms are too large. Its failure to print anything completes the proof.

$\mathbb{Z}[\varphi]$-irreducible element dividing p.

Proof. If p is reducible in $\mathbb{Z}[\varphi]$ then there exist non-units $u, v \in \mathbb{Z}[\varphi]$ with $uv = p$ and accordingly $N(u) \mid N(p) = p^2$; that is, $N(u) = p$.

- Observe that $a^2 + ab - b^2 \equiv (a - 2b)^2 \pmod 5$ and that neither of ± 2 are quadratic residues modulo 5. This proves the first part of the proposition, as if $u = a + b\varphi$ then $N(u)$ can never equal p.

- To prove the second part, we first show that if $p \equiv 1 \pmod 5$ then there exists $x \in \mathbb{Z}$ satisfying $x^2 - x - 1 \equiv 0 \pmod p$. Indeed, rewrite the equation to $(x - 2^{-1})^2 \equiv 1 + 4^{-1}$. Now,
$$\left(\frac{1+4^{-1}}{p}\right) = \left(\frac{1+4^{-1}}{p}\right)\left(\frac{4}{p}\right) = \left(\frac{5}{p}\right) = \left(\frac{p}{5}\right)(-1)^{p-1} = \left(\frac{p}{5}\right) = 1$$
with the third equality following by quadratic reciprocity. Let $y \in \mathbb{Z}$ be any modulo-p square root of $1 + 4^{-1}$. Then $x = y + 2^{-1}$ suffices. Set $z = x - \varphi \in \mathbb{Z}[\varphi]$. Then $p \mid x^2 - x - 1 = zz^\bullet$ but p does not divide z as z's φ-part is not divisible by p, so p is not irreducible. Thus $u = \gcd(p, z)$ will be a non-unit divisor of p, as desired. (u cannot be a unit because otherwise $\gcd(p^\bullet, z^\bullet) = \gcd(p, z^\bullet)$ will also be a unit, but then there are no prime divisors of $zz^\bullet$ also dividing p, a contradiction.)

  The timing of the above method is due to the extended Euclidean algorithm for modular inverses, and the standard Euclidean algorithm for GCDs in a Euclidean domain, which have respective runtimes $O(\log^2 p)$ and $O(\log p)$.

□

The only case not covered in the above is $p = 5$, which trivially factors as $(-1 + 2\varphi)^2$.

Algorithm 2. *Assume that there is an efficient blackbox algorithm to factor $n \in \mathbb{Z}$ in time $O(\text{poly}\log n)$. Then there is an algorithm which factors $n \in \mathbb{Z}[\varphi]$ in time $O(\text{poly}\log|N(n)|)$.*

Proof. Factor $N(n)$ as $\prod_{\ell=1}^{m} p_\ell^{e_\ell}$. Each prime p_ℓ is factorizable in time $O(\text{poly}\log p_\ell) \subset O(\text{poly}\log N(n))$, by Proposition 2. There are $O(\log N(n))$ such primes. □

To each irreducible $u \in \mathbb{Z}[\varphi]$, let its *associated prime* be the least (positive) prime factor of $N(u)$; that is, $\sqrt{N(u)}$ when u is a $\mathbb{Z}$-irreducible times a unit, and $N(u)$ otherwise.[d]

Lemma 1. *For any $u \in \mathbb{Z}[\varphi] \backslash \{0\}$, it is not the case that both u and $u\varphi$ can be written as a sum of two squares.*

Proof. Suppose not, so write $u = w^2 + x^2$ and $u\varphi = y^2 + z^2$. Then we may take the product

$$u^\bullet u\varphi = N(u)\varphi = (w^\bullet y + x^\bullet z)^2 + (w^\bullet z - x^\bullet y)^2 = (a + b\varphi)^2 + (c + d\varphi)^2$$

for some $a, b, c, d \in \mathbb{Z}$. The right-hand side expands out with 1-part[e]equal to $a^2 + b^2 + c^2 + d^2$, but $N(u)\varphi$ has 1-part equal to 0, so $a = b = c = d = 0$, and therefore $N(u) = 0$, a contradiction. □

[d] Let us quickly establish why the associated prime is well-defined. Suppose $p \neq q$ are $\mathbb{Z}$-primes with $p, q \mid N(u) = uu^\bullet$. Let $p', q' \in \mathbb{Z}[\varphi]$ be irreducibles such that $|N(p')| = p$ if $p \equiv \pm 1 \pmod 5$ or $p = 5$, and $p' = p$ otherwise; and define q' similarly. Then $p', q' \mid uu^\bullet$, so as irreducibles we have p' dividing either u or $u^\bullet$, and similarly for q', hence $p = q$ since u and $u^\bullet$ are themselves irreducible. Thus, $N(u)$ is divisible by at most one prime at most twice.

[e] Viewing $\mathbb{Q}(\varphi) \cong \langle 1, \varphi \rangle_{\mathbb{Q}} \cong \mathbb{Q}^2$ so that the "1-part" and "φ-part" are the corresponding components in the canonical isomorphism.

Proposition 3. *Let $u \in \mathbb{Z}[\varphi]$ be irreducible, with associated prime p. If $p \equiv 1, 3, 7, 9, 13, 17 \pmod{20}$ then there is an efficient algorithm in time $O(\operatorname{poly}\log p)$ to write either u or $u\varphi$ as a sum of two squares.*

Observe that $\phi(20) = 8$ (ϕ here signifying Euler's totient function), so by Chebotarev's density theorem the above-asserted algorithm manages to write $3/4$ of primes as sums of two squares.

Proof. Before beginning in earnest, first remark that 2 factors in $\mathbb{Z}[i,\varphi]$ as $(1+i)(1-i)$, and that $1 \pm i$ are irreducible because $N_{\mathbb{Q}(i,\varphi)/\mathbb{Q}(\varphi)}(1 \pm i) = 2$, a $\mathbb{Z}[\varphi]$-irreducible.

- Suppose $p \equiv 1 \pmod 4$ (i.e. $p \equiv 1, 9, 13, 17 \pmod{20}$). Then there exists even $x \in \mathbb{Z}$ satisfying $x^2 + 1 \equiv 0 \pmod p$, that is, $u \mid p \mid x^2 + 1$. Let $g = \gcd(u, x+i)$, computed in $\mathbb{Z}[i,\varphi]$, and set $s = \operatorname{Re}(g)$ and $t = \operatorname{Im}(g)$. We claim that either u or $u\varphi$ equals a squared unit times $s^2 + t^2$. Indeed, suppose $q \in \mathbb{Z}[i,\varphi]$ is an irreducible divisor of u. We wish to show that q divides exactly one of $x \pm i$. (Clearly it divides at least one of them, by $q \mid u \mid x^2 + 1$.)
  - Say $1 \pm i \neq q \in \mathbb{Z}[i,\varphi]$ is an irreducible dividing u and both $x \pm i$. Then $q \mid 2i$. By our choice of q this is impossible.
  - Suppose now we let q be one of $1 \pm i$. Then in order to have $1 \pm_1 i \mid x \pm_2 i$ (where $\pm_1$ and $\pm_2$ are independent signs) we must have $(1 \pm_1 i)(a + bi) = x \pm_2 i$ where $a, b \in \mathbb{Z}[\varphi]$. Solving for a and b by looking at real and imaginary parts, we have

    $$\begin{aligned} a \mp_1 b &= \quad x \\ b \pm_1 a &= \pm_2 1 \end{aligned}$$

    and solving yields $2a = x \pm_1 \pm_2 1$, so in order for a to exist we must have x is odd. This is a contradiction to our assumption that x is even, regardless of the choice of $\pm_1$ and $\pm_2$, thus in fact we have neither of $1 \pm i$ dividing either of $x \pm i$.

  Thus, if q divides u with multiplicity m_q (so that u factors in $\mathbb{Z}[i,\varphi]$ as $u = \varphi^m \prod_{q \mid u} q^{m_q}$), we must have that q divides $x \pm i$ with multiplicity at least m and $x \mp i$ with multiplicity 0; and so[f] $g = \prod_{q \mid u, x+i} q^{m_q}$ while $\bar{g} = \gcd(u, x - i) = \prod_{q \mid u, x-i} q^{m_q}$. Thus $u = \varphi^m g\bar{g} = \varphi^m(\operatorname{Re}(g)^2 + \operatorname{Im}(g)^2)$. If $2 \mid m$ then $u = (\varphi^{m/2}\operatorname{Re}(g))^2 + (\varphi^{m/2}\operatorname{Im}(g))^2$.

- Suppose $p \equiv 3 \pmod 4, \pm 2 \pmod 5$ (i.e. $p \equiv 3, 7 \pmod{20}$). Then compute

  $$\left(\frac{-5}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{5}{p}\right) = -\left(\frac{p}{5}\right)(-1)^{p-1} = 1$$

  so that there exists even $x \in \mathbb{Z}$ which is not a multiple of 5 satisfying $x^2 + 5 \equiv 0 \pmod p$, that is, $u = p \mid x^2 + 5$. Let $g = \gcd(g, x + i(-1 + 2\varphi))$, computed in $\mathbb{Z}[i,\varphi]$, and set $s = \operatorname{Re}(g)$ and $t = \operatorname{Im}(g)$. We claim that $p = s^2 + t^2$. Indeed, suppose $q \in \mathbb{Z}[i,\varphi]$ is an irreducible divisor of p (in $\mathbb{Z}[i,\varphi]$). We wish to show that q divides exactly one of $x \pm i(-1 + 2\varphi)$. (Clearly it divides at least one of them, by $q \mid p \mid x^2 + 5$.)

[f] Depending on choice of GCD algorithm, there may be a unit factor in front, but we assume not without loss of generality.

- Say $-1+2\varphi, 1\pm i \neq q \in \mathbb{Z}[i,\varphi]$ is an irreducible dividing u and both $x\pm i(-1+2\varphi)$. Then $q \mid 2(-1+2\varphi)i$. By our choice of q this is impossible.
- Suppose we let $q = -1+2\varphi = \sqrt{5}$. Then as 5 does not divide x, we have

$$\frac{x \pm i(-1+2\varphi)}{q} = \frac{x}{5}(-1+2\varphi) \pm i \in \mathbb{Q}(i,\varphi)\backslash\mathbb{Z}[i,\varphi]$$

  and so in fact q divides neither of $x \pm i(-1+2\varphi)$.
- Suppose now we let q be one of $1 \pm i$. Then in order to have $1 \pm_1 i \mid x \pm_2 i(-1+2\varphi)$ (where $\pm_1$ and $\pm_2$ are independent signs) we must have $(1 \pm_1 i)(a+bi) = x \pm_2 i(-1+2\varphi)$ where $a, b \in \mathbb{Z}[\varphi]$. Solving for a and b by looking at real and imaginary parts, we have

$$\begin{aligned} a \mp_1 b &= x \\ b \pm_1 a &= \pm_2(-1+2\varphi) \end{aligned}$$

  and solving yields $2a = x \pm_1 \pm_2(-1+2\varphi)$, so in order for a to exist we must have x is odd. This is a contradiction to our assumption that x is even, regardless of the choice of $\pm_1$ and $\pm_2$, thus in fact we have neither of $1 \pm i$ dividing either of $x \pm i$.

It is here that we require Theorem 2, that $\mathbb{Z}[i,\varphi]$ is norm-Euclidean, as this ensures that GCDs are efficiently computable, using an Euclidean algorithm with respect to the Euclidean function (the norm). As before, the bottlenecks are in the extended and standard Euclidean algorithms, which are both of runtime $O(\log^2 p)$. □

Observe that we can also include the $\mathbb{Z}$-irreducibles $2 = 1^2 + 1^2$ and $5 = (-1+2\varphi)^2 + 0^2$.

Theorem 3. *For $x \in \mathbb{Z}[\varphi]$, factor it as*

$$x = \prod_{\ell=1}^{n} u_\ell^{m_\ell}$$

where $u_\ell \in \mathbb{Z}[\varphi]$ are all irreducible, with associated primes p_ℓ, and $m_\ell \in \mathbb{N}$. Then either x or $x\varphi$ may be written as the sum of two squares if, for all $\ell \in [n]$, one of the following holds:

- *$p_\ell = 2$;*
- *$p_\ell \equiv 1, 3, 7, 9, 13, 17 \pmod{20}$;*
- *$2 \mid m_\ell$;*

and, for arbitrary x not a priori satisfying all three criteria, this whole procedure (i.e., either determining a sum of two squares, or rejecting on the basis of some ℓ failing all three conditions) runs in time $O(\text{poly}\log|N(x)|)$.

Proof. For each $\ell \in [n]$, consider s_ℓ and t_ℓ defined as follows:

- If $p_\ell = 2$ then $s_\ell = t_\ell = 1$.
- If $p_\ell \equiv 1, 3, 7, 9, 13, 17 \pmod{20}$ then s_ℓ and t_ℓ are as computed using Proposition 3.
- If $2 \mid m_\ell$ then $s_\ell = u_\ell^{m_\ell/2}$ and $t_\ell = 0$.

Now, we do the standard (inductive) trick where $(s_\ell^2 + t_\ell^2)(s_{\ell'}^2 + t_{\ell'}^2) = (s_\ell s_{\ell'} + t_\ell t_{\ell'})^2 + (s_\ell t_{\ell'} - t_\ell s_{\ell'})^2$. Since we have that $s_\ell^2 + t_\ell^2$ equals either u_ℓ or $u_\ell \varphi$, the resulting product from performing the trick n times gives a sum of two squares $s^2 + t^2$ equal to $x\varphi^{n'}$ where $0 \leq n' \leq n$. If $2 \mid n'$ then we return the pair $(s\varphi^{-n'/2}, t\varphi^{-n'/2})$, and otherwise we return $(s\varphi^{(1-n')/2}, t\varphi^{(1-n')/2})$.

For the runtime analysis, simply factor x using Algorithm 2 and then for each resulting factor, apply Proposition 3 after checking that one of the three criteria holds (immediately halting and rejecting if any factor fails). □