Universality and Solovay-Kitaev Theorem

Seminar Quantum Algorithms

Alexander Kulpe

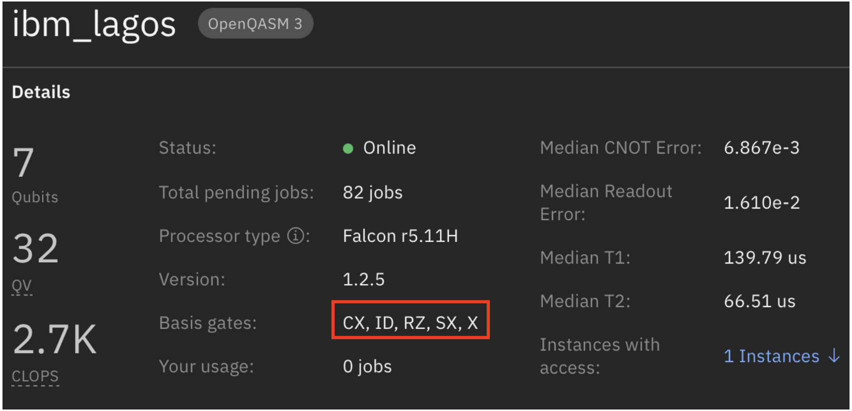
Ruhr-University Bochum
University of Cologne

April 25, 2023

Table of Contents

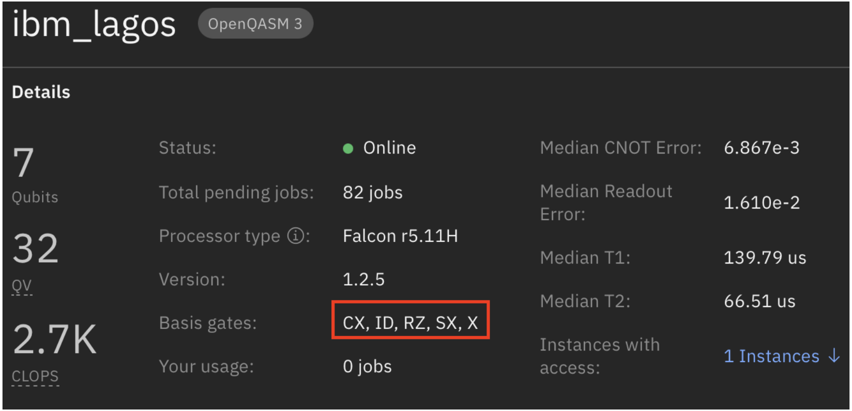
Table of Contents

Motivation



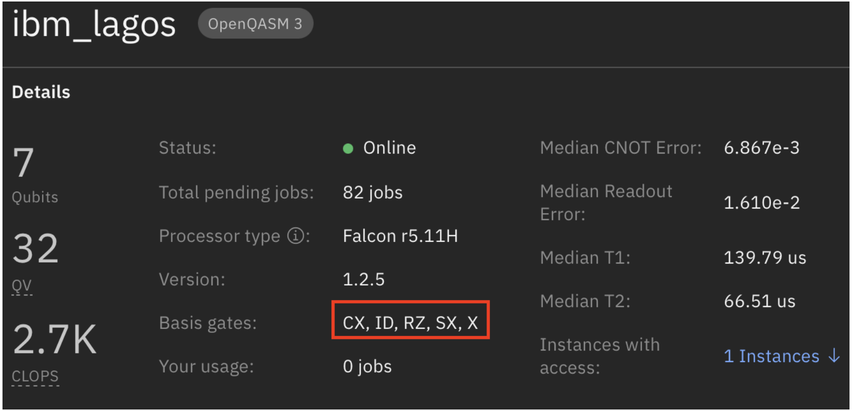
Can we compute Quantum Circuits with small set of Basis gates?

Motivation



Can we compute Quantum Circuits with small set of Basis gates?
Can we compute efficiently with this set of Basis gates?

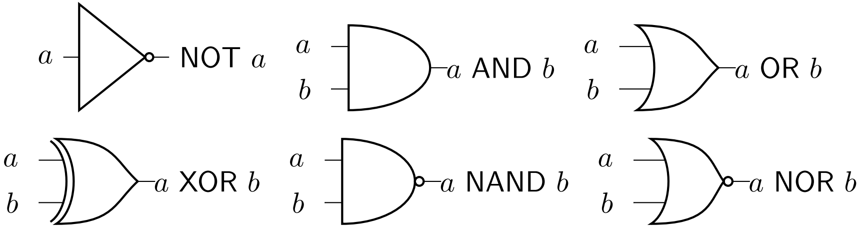
Motivation



Can we compute Quantum Circuits with small set of Basis gates?
Can we compute efficiently with this set of Basis gates?
Is the complexity of quantum algorithms dependent on supported Basis gates?

Table of Contents

Elementary gates



Every gate that we can think of can be described by a truth table

Universality

- Claim: There exists a universal gate set s.t. we can compute every function $f : \{0,1\}^n \rightarrow \{0,1\}^m \Rightarrow$ Proof by induction

Universality

- Claim: There exists a universal gate set s.t. we can compute every function $f : \{0,1\}^n \to \{0,1\}^m \Rightarrow$ Proof by induction
- Consider $f : \{0,1\}^n \to \{0,1\}$

Universality

- Claim: There exists a universal gate set s.t. we can compute every function $f : \{0,1\}^n \to \{0,1\}^m \Rightarrow$ Proof by induction
- Consider $f : \{0,1\}^n \to \{0,1\}$
- $n = 1$: Four possible functions (truth table)

Universality

- Claim: There exists a universal gate set s.t. we can compute every function $f : \{0,1\}^n \to \{0,1\}^m \Rightarrow$ Proof by induction
- Consider $f : \{0,1\}^n \to \{0,1\}$
- $n = 1$: Four possible functions (truth table)
- $f_0(x_1, \ldots, x_n) \equiv f(0, x_1, \ldots, x_n), f_1(x_1, \ldots, x_n) \equiv f(1, x_1, \ldots, x_n)$
- $f(x) = (\overline{x_0} \cdot f_0(x_1, \ldots, x_n)) \oplus (x_0 \cdot f_1(x_1, \ldots, x_n))$

Universality

- Claim: There exists a universal gate set s.t. we can compute every function $f : \{0,1\}^n \to \{0,1\}^m \Rightarrow$ Proof by induction
- Consider $f : \{0,1\}^n \to \{0,1\}$
- $n = 1$: Four possible functions (truth table)
- $f_0(x_1, \ldots, x_n) \equiv f(0, x_1, \ldots, x_n), f_1(x_1, \ldots, x_n) \equiv f(1, x_1, \ldots, x_n)$
- $f(x) = (\overline{x_0} \cdot f_0(x_1, \ldots, x_n)) \oplus (x_0 \cdot f_1(x_1, \ldots, x_n))$
- NAND gate is universal (exercise)

Table of Contents

Goal

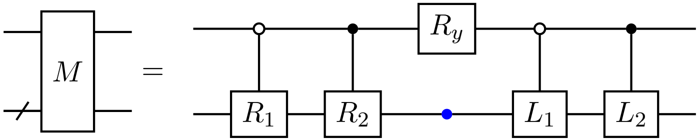
Express an arbitrary n-Qubit gate with a sequence of 1-Qubit gates and CNOT

Goal

Express an arbitrary n-Qubit gate with a sequence of 1-Qubit gates and CNOT

Arbitrary Unitary Gates

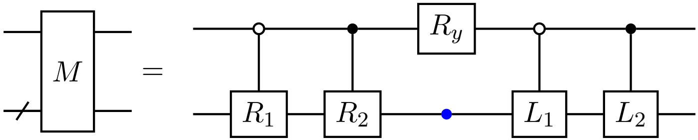
- Fact: M can be represented as $M = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \cdot \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \cdot \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}$ where C, S are diagonal matrices with real entries and $C^2 + S^2 = I$



- Uniformly Controlled Rotation can be implemented with CNOT and rotation gates

Arbitrary Unitary Gates

- Fact: M can be represented as $M = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} \cdot \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \cdot \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}$ where C, S are diagonal matrices with real entries and $C^2 + S^2 = I$



- Uniformly Controlled Rotation can be implemented with CNOT and rotation gates

$\Rightarrow$ n-Qubit gates can be expressed as a sequence of controlled gates, CNOT gates, and rotation gates

Goal

Express controlled gates as a sequence of 1-Qubit gates and CNOT

1-Qubit Gates

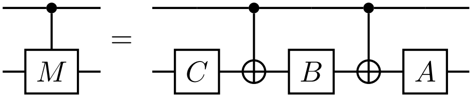
$$\mathrm{SU}(2) = \left\{ \begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} \mid \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1 \right\}$$

- arbitrary unitary 2×2 matrix only differs by global phase shift (exercise)
- every matrix $M \in \mathrm{SU}(2)$ can be represented as $M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$

Controlled Gates

Lemma

For any $M \in \mathrm{SU}(2)$, *there exist matrices* A, B, C *s.t.* $A \cdot B \cdot C = I$ *and* $A \cdot X \cdot B \cdot X \cdot C = M$.



- for arbitrary unitary 2×2 matrix additional controlled phase gate (relative phase shift)

Controlled Gates - Proof

$$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$A \cdot B \cdot C = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot R_z\left(\frac{\beta-\alpha}{2}\right)$$

Controlled Gates - Proof

$$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$\begin{aligned} A \cdot B \cdot C &= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\ &= R_z(\alpha) \cdot R_z(-\alpha) \end{aligned}$$

Controlled Gates - Proof

$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$
\begin{aligned}
A \cdot B \cdot C &= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_z(-\alpha) \\
&= I
\end{aligned}
$$

Controlled Gates - Proof

$$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$AXBXC = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right)$$

Controlled Gates - Proof

$$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$\begin{aligned}
AXBXC &= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot X \cdot X \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right)
\end{aligned}$$

Controlled Gates - Proof

$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$\begin{aligned}
AXBXC &= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot X \cdot X \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_z\left(\frac{\alpha+\beta}{2}\right) \cdot R_z\left(\frac{\beta-\alpha}{2}\right)
\end{aligned}$$

Controlled Gates - Proof

$$M = R_z(\alpha)\cdot R_y(\theta)\cdot R_z(\beta)$$

Set $A = R_z(\alpha)\cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right)\cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$\begin{aligned}
AXBXC &= R_z(\alpha)\cdot R_y\left(\frac{\theta}{2}\right)\cdot X\cdot R_y\left(-\frac{\theta}{2}\right)\cdot R_z\left(-\frac{\alpha+\beta}{2}\right)\cdot X\cdot R_z\left(\frac{\beta-\alpha}{2}\right)\\
&= R_z(\alpha)\cdot R_y\left(\frac{\theta}{2}\right)\cdot X\cdot R_y\left(-\frac{\theta}{2}\right)\cdot X\cdot X\cdot R_z\left(-\frac{\alpha+\beta}{2}\right)\cdot X\cdot R_z\left(\frac{\beta-\alpha}{2}\right)\\
&= R_z(\alpha)\cdot R_y\left(\frac{\theta}{2}\right)\cdot R_y\left(\frac{\theta}{2}\right)\cdot R_z\left(\frac{\alpha+\beta}{2}\right)\cdot R_z\left(\frac{\beta-\alpha}{2}\right)\\
&= R_z(\alpha)\cdot R_y(\theta)\cdot R_z(\beta)
\end{aligned}$$

Controlled Gates - Proof

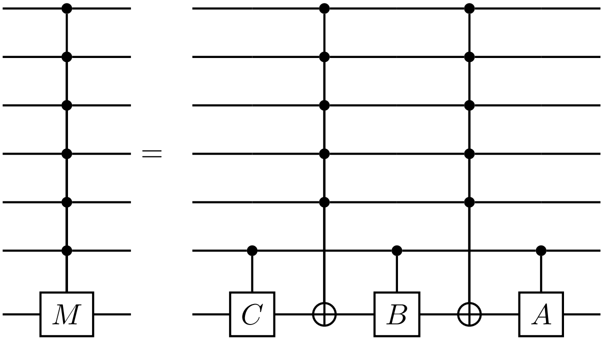
$M = R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta)$

Set $A = R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right)$, $B = R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right)$, $C = R_z\left(\frac{\beta-\alpha}{2}\right)$.

$$\begin{aligned}
AXBXC &= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot X \cdot R_y\left(-\frac{\theta}{2}\right) \cdot X \cdot X \cdot R_z\left(-\frac{\alpha+\beta}{2}\right) \cdot X \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_y\left(\frac{\theta}{2}\right) \cdot R_z\left(\frac{\alpha+\beta}{2}\right) \cdot R_z\left(\frac{\beta-\alpha}{2}\right) \\
&= R_z(\alpha) \cdot R_y(\theta) \cdot R_z(\beta) \\
&= M
\end{aligned}$$

Controlled Gates II

$M \in \mathrm{SU}(2)$

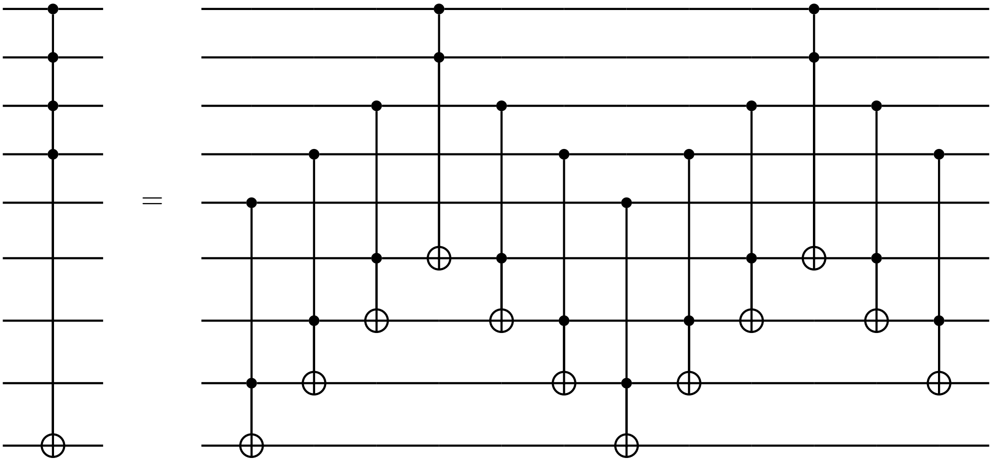


$\Rightarrow$ Multiple-Controlled gates can be realized with Multiple-Controlled Toffoli gates

Goal

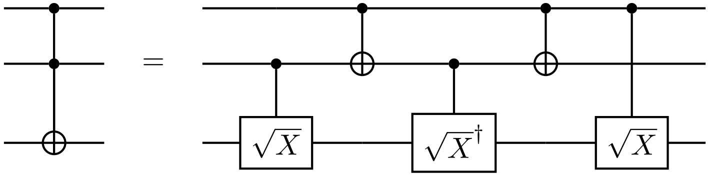
Express Multiple-Controlled Toffoli gates as a sequence of 1-Qubit gates and CNOT

Multiple-Controlled Toffoli gates



$\Rightarrow$ Multiple-Controlled Toffoli gates can be realized with Toffoli gates

Toffoli gates



$\Rightarrow$ Toffoli gate can be realized with controlled 1-Qubit gates and CNOT. Controlled 1-Qubit gates can be realized using previous lemma.

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD
- Controlled Gates $\Rightarrow$ Toffoli gates and Single-Controlled gates

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD
- Controlled Gates $\Rightarrow$ Toffoli gates and Single-Controlled gates
- Toffoli gate $\Rightarrow$ Single-Controlled gates and CNOT
- Single-Controlled gate $\Rightarrow$ 1-Qubit gates and CNOT via Lemma

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD
- Controlled Gates $\Rightarrow$ Toffoli gates and Single-Controlled gates
- Toffoli gate $\Rightarrow$ Single-Controlled gates and CNOT
- Single-Controlled gate $\Rightarrow$ 1-Qubit gates and CNOT via Lemma
- quantum circuits can be implemented exactly
- But: Discrete Universal Gate Set more practical ($H, \mathrm{Ph}, \mathrm{CNOT}, T$ are universal)

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD
- Controlled Gates $\Rightarrow$ Toffoli gates and Single-Controlled gates
- Toffoli gate $\Rightarrow$ Single-Controlled gates and CNOT
- Single-Controlled gate $\Rightarrow$ 1-Qubit gates and CNOT via Lemma
- quantum circuits can be implemented exactly
- But: Discrete Universal Gate Set more practical ($H, \mathrm{Ph}, \mathrm{CNOT}, T$ are universal)
- Question: Can we efficiently approximate quantum circuits?

Quick Recap

- Arbitray Unitary $\Rightarrow$ Controlled gates, 1-Qubit gates, and CNOT via CSD
- Controlled Gates $\Rightarrow$ Toffoli gates and Single-Controlled gates
- Toffoli gate $\Rightarrow$ Single-Controlled gates and CNOT
- Single-Controlled gate $\Rightarrow$ 1-Qubit gates and CNOT via Lemma
- quantum circuits can be implemented exactly
- But: Discrete Universal Gate Set more practical ($H, \mathrm{Ph}, \mathrm{CNOT}, T$ are universal)
- Question: Can we efficiently approximate quantum circuits?

$\Rightarrow$ Solovay-Kitaev

Table of Contents

Informal

Given an appropriate subset of $\mathrm{SU}(2)$, we can efficiently approximate every possible element in $\mathrm{SU}(2)$ arbitrarily well.

History Overview

1995 Solovay announces the $\mathrm{SU}(2)$ result over an email list

1997 Kitaev publishes result for $\mathrm{SU}(d)$ with algorithm

History Overview

- **1995** Solovay announces the $\mathrm{SU}(2)$ result over an email list
- **1997** Kitaev publishes result for $\mathrm{SU}(d)$ with algorithm
- **2000** During a talk, Solovay says that "to my great sorrow, I have to use the inverses". The lecture is interrupted by a fire alarm.
- **2010s** Results on most efficient compilation for specific sets
- **2016** Sardharwalla, Cubitt, Harrow, Linden show how Pauli group can be used to produce approximate inverses.
- **2017** Bouland, Ozols: Property can be generalized to any gate set which contains an irreducible representation of a finite group.

History Overview

1995 Solovay announces the $\mathrm{SU}(2)$ result over an email list

1997 Kitaev publishes result for $\mathrm{SU}(d)$ with algorithm

2000 During a talk, Solovay says that "to my great sorrow, I have to use the inverses". The lecture is interrupted by a fire alarm.

2010s Results on most efficient compilation for specific sets

2016 Sardharwalla, Cubitt, Harrow, Linden show how Pauli group can be used to produce approximate inverses.

2017 Bouland, Ozols: Property can be generalized to any gate set which contains an irreducible representation of a finite group.

2020 Oszmaniec, Sawicki, Horodecki: Non-constructive inverse-free Solovay-Kitaev using results about spectral gaps of random walks on compact groups

2021 Bouland, Giurgica-Tiron: Constructive inverse-free Solovay-Kitaev

Informal

Given an appropriate subset of $\mathrm{SU}(2)$, we can efficiently approximate every possible element in $\mathrm{SU}(2)$ arbitrarily well.

Useful definitions - metric spaces

Let (X, d) be a metric space.

Definition

Let $A, N \subset X$ where N ist finite and $\varepsilon > 0$. N is called ε-net for A if

$$\forall a \in A \; \exists p \in N : d(a, p) < \varepsilon$$

Example

$\{0, 1\}$ is a $2/3$-net for the interval $[0, 1]$ but not for the interval $[0, 2]$.

Useful definitions - metric spaces

Let (X, d) be a metric space.

Definition

Let $A, N \subset X$ where N ist finite and $\varepsilon > 0$. N is called ε-net for A if

$$\forall a \in A \; \exists p \in N : d(a, p) < \varepsilon$$

Example

$\{0, 1\}$ is a $2/3$-net for the interval $[0, 1]$ but not for the interval $[0, 2]$.

Definition

$D \subset X$ is dense in X if

$$\forall x \in X \; \forall \varepsilon > 0 \; \exists p \in D : d(x, p) < \varepsilon$$

Example

$\mathbb{Q}$ is dense in $\mathbb{R}$. $\mathbb{N}$ is not dense in $\mathbb{R}$.

Useful definitions - metric spaces

Let (X, d) be a metric space.

Definition

Let $A, N \subset X$ where N ist finite and $\varepsilon > 0$. N is called ε-net for A if

$$\forall a \in A \, \exists p \in N : d(a, p) < \varepsilon$$

Example

$\{0, 1\}$ is a $2/3$-net for the interval $[0, 1]$ but not for the interval $[0, 2]$.

Definition

$D \subset X$ is dense in X if

$$\forall x \in X \, \forall \varepsilon > 0 \, \exists p \in D : d(x, p) < \varepsilon$$

Example

$\mathbb{Q}$ is dense in $\mathbb{R}$. $\mathbb{N}$ is not dense in $\mathbb{R}$.

Useful definitions - trace norm

Definition

$$||A|| := \mathrm{tr}\,|A| = \mathrm{tr}\sqrt{A^\dagger A}$$

is called the trace norm.

The metric induced by the trace norm is given by $d(A,B) := ||A-B||$ and satisfies the following properties:

- unitary invariance: $||UAV|| = ||A||$ for any unitaries U and V,
- triangle inequality: $||A+B|| \leq ||A|| + ||B||$,
- submultiplicativity: $||AB|| \leq ||A|| \cdot ||B||$

Informal

Given an appropriate subset of $\mathrm{SU}(2)$, we can efficiently approximate every possible element in $\mathrm{SU}(2)$ arbitrarily well.

Gate set

- Let $\mathcal{G} \subset \mathrm{SU}(2)$ be a gate set.
- For the proof of Solovay-Kitaev we need $\mathcal{G}$ to be closed under inverses or do we?
- Notation: $\mathcal{G}^\ell = \left\{ g_1^{\alpha_1} g_2^{\alpha_2} \dots g_\ell^{\alpha_\ell} \mid g_i \in \mathcal{G}, \alpha_i = \pm 1 \right\}$, $\langle \mathcal{G} \rangle := \mathcal{G}^0 \cup \mathcal{G}^1 \cup \mathcal{G}^2 \cup \dots$

Gate set

- Let $\mathcal{G} \subset \mathrm{SU}(2)$ be a gate set.
- For the proof of Solovay-Kitaev we need $\mathcal{G}$ to be closed under inverses or do we?
- Notation: $\mathcal{G}^\ell = \left\{ g_1^{\alpha_1} g_2^{\alpha_2} \dots g_\ell^{\alpha_\ell} \mid g_i \in \mathcal{G}, \alpha_i = \pm 1 \right\}$, $\langle \mathcal{G} \rangle := \mathcal{G}^0 \cup \mathcal{G}^1 \cup \mathcal{G}^2 \cup \dots$
- Solovay-Kitaev: We assume that $\mathcal{G}$ is finite subset of $\mathrm{SU}(2)$ that is closed under inverses and $\langle \mathcal{G} \rangle$ is dense in $\mathrm{SU}(2)$.

Solovay-Kitaev Theorem

Theorem

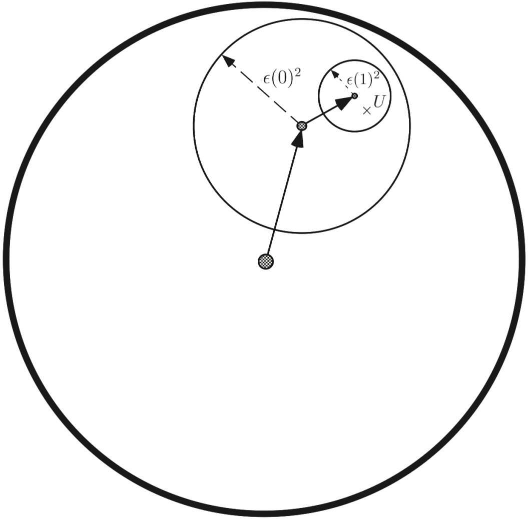
There is a constant c *s.t. for any* $\mathcal{G}$ *that is closed under inverses and* $\langle\mathcal{G}\rangle$ *is dense in* $\mathrm{SU}(2)$ *and* $\varepsilon > 0$ *one can choose* $\ell = \mathcal{O}(\log^c(1/\varepsilon))$ *so that* $\mathcal{G}^\ell$ *is an* ε*-net for* $\mathrm{SU}(2)$*. Furthermore, there exists an efficient algorithm that finds this approximation.*

In other words: The overhead of computing with a discrete universal gate set is poly-logarithmic.

Algorithm - Idea

Let $S_\varepsilon := \{U \in \mathrm{SU}(2) \mid ||U - I|| < \varepsilon\}$ be an open ε-ball in $\mathrm{SU}(2)$ around the identity

Construct series of ε-nets $\Gamma_0, \Gamma_1, \ldots$ s.t.

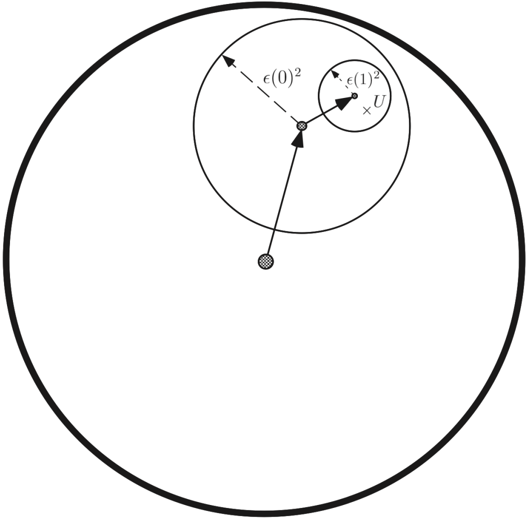


Algorithm - Idea

Let $S_\varepsilon := \{U \in \mathrm{SU}(2) \mid ||U - I|| < \varepsilon\}$ be an open ε-ball in $\mathrm{SU}(2)$ around the identity

Construct series of ε-nets $\Gamma_0, \Gamma_1, \ldots$ s.t.

- Γ_0 is $\varepsilon(0)^2$-net for $\mathrm{SU}(2)$ and
- Γ_k is $\varepsilon(k)^2$-net for $S_{\varepsilon(k)}$ for $k > 0$.



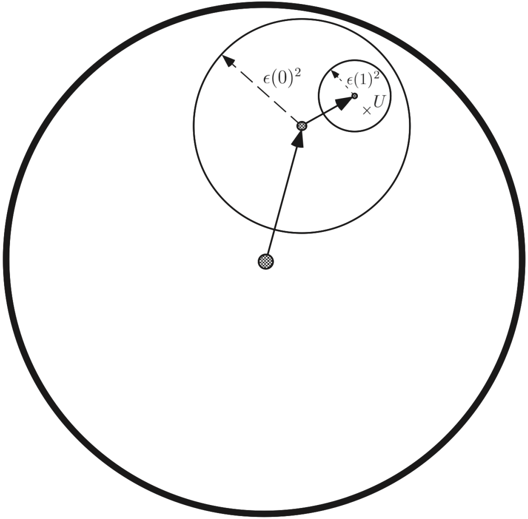
Algorithm - Idea

Let $S_\varepsilon := \{U \in \mathrm{SU}(2) \mid ||U - I|| < \varepsilon\}$ be an open ε-ball in $\mathrm{SU}(2)$ around the identity

Construct series of ε-nets $\Gamma_0, \Gamma_1, \ldots$ s.t.

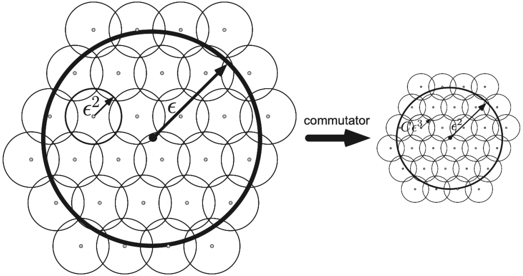
- Γ_0 is $\varepsilon(0)^2$-net for $\mathrm{SU}(2)$ and
- Γ_k is $\varepsilon(k)^2$-net for $S_{\varepsilon(k)}$ for $k > 0$.

1. Start with initial approximation
2. Attack remaining distance with techniques that rely on being near the identity
3. Express precise matrices near the identity as strings of less precise matrices that are farther from the identity



Algorithm - Idea II

- Initial net Γ_0 can be created in constant time



Algorithm - Idea II

- Initial net Γ_0 can be created in constant time
- recursively: $\Gamma_k = [\![\Gamma_{k-1}, \Gamma_{k-1}]\!] := \{[\![A, B]\!] \mid A, B \in \Gamma_{k-1}\}$ where $[\![A, B]\!] = ABA^\dagger B^\dagger$ denotes the group commutator

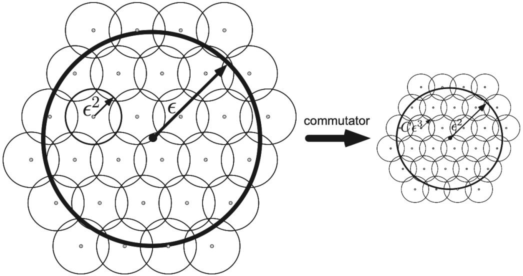


Figure: Taking group commutator of elements in S_ε fills in S_{ε^2} much more densely (Shrinking Lemma)

Shrinking Lemma

Lemma

There exist ε', s s.t. for any $\mathcal{G}$ and $\varepsilon \leq \varepsilon'$ we have: If $\mathcal{G}^{\ell}$ is an ε^2-net for S_ε then $\mathcal{G}^{5\ell}$ is an $s\varepsilon^3$-net for $S_{\sqrt{s\varepsilon^3}}$

Shrinking Lemma

Lemma
There exist ε', s s.t. for any $\mathcal{G}$ and $\varepsilon \leq \varepsilon'$ we have: If $\mathcal{G}^{\ell}$ is an ε^2-net for S_{ε} then $\mathcal{G}^{5\ell}$ is an $s\varepsilon^3$-net for $S_{\sqrt{s\varepsilon^3}}$

Corollary
There exist ε', s s.t. for any $\mathcal{G}, \varepsilon_0 \leq \varepsilon'$, and $k \in \mathbb{N}$ we have: If $\mathcal{G}^{\ell_0}$ is an ε^2-net for S_{ε_0} then $\mathcal{G}^{\ell_k}$ is an ε_k^2-net for S_{ε_k} where $\ell_k := 5^k \ell_0$ and $\varepsilon_k := (s\varepsilon_0)^{(3/2)^k}/s$.

Proof Solovay-Kitaev Idea

Theorem

There is a constant c s.t. for any $\mathcal{G}$ and $\varepsilon > 0$ one can choose $\ell = \mathcal{O}(\log^c(1/\varepsilon))$ so that $\mathcal{G}^\ell$ is an ε-net for $SU(2)$.

Corollary

There exist ε', s s.t. for any $\mathcal{G}, \varepsilon_0 \leq \varepsilon'$, and $k \in \mathbb{N}$ we have: If $\mathcal{G}^{\ell_0}$ is an ε^2-net for S_{ε_0} then $\mathcal{G}^{\ell_k}$ is an ε_k^2-net for S_{ε_k} where $\ell_k := 5^k \ell_0$ and $\varepsilon_k := (s\varepsilon_0)^{(3/2)^k}/s$.

The corollary allows to obtain good approximation for any element of $\mathrm{SU}(2)$ that is sufficiently close to identity. We now have to obtain a good approximation for any element of $\mathrm{SU}(2)$.
Start with rough approximation and use shrinking lemma.

Proof Solovay-Kitaev / Algorithm

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

SK(U,n)

Input: $U \in \mathrm{SU}(2)$, depth n

Ouptut: $V \in \langle \mathcal{G} \rangle$ s.t. $||U - V|| < \varepsilon^2(n)$

if $n = 0$ **do**

$V = \varepsilon^2(0) - \text{APPROX}(U, G_I)$

else

$W = SK(U, n-1)$

$A, B = \text{FACTOR}(UW^\dagger)$

$V = [\![\text{SK}(A, n-1), \text{SK}(B, n-1)]\!] W$

Proof Solovay-Kitaev: Step 1

Choose ε_0 s.t.

- $\varepsilon_0 < \varepsilon'$ so that we can use Shrinking lemma
- $s\varepsilon_0 < 1$ so that (ε_k) decreases
- ε_0 small s.t. $\varepsilon_k^2 < \varepsilon_{k+1}$ so we can find closest current approximaton to our gate

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 2

$\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ we can find ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 2

$\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ we can find ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$
Given $U \in \mathrm{SU}(2)$ we can choose $U_0 \in \mathcal{G}^{\ell_0}$ s.t. $||U - U_0|| < \varepsilon_0^2$.

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 2

$\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ we can find ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$
Given $U \in \mathrm{SU}(2)$ we can choose $U_0 \in \mathcal{G}^{\ell_0}$ s.t. $||U - U_0|| < \varepsilon_0^2$.

Define $\Delta_1 := UU_0^\dagger$. Then:

$$||\Delta_1 - I|| = \left|\left|(U - U_0)U_0^\dagger\right|\right| = ||U - U_0|| < \varepsilon_0^2 < \varepsilon_1$$

$\Rightarrow \Delta_1 \in S_{\varepsilon_1}$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 3

Shrinking Lemma $\Rightarrow \exists U_1 \in \mathcal{G}^{\ell_1}$ s.t.

$$||\Delta_1 - U_1|| = \left|\left|UU_0^\dagger - U_1\right|\right| = ||U - U_1U_0|| < \varepsilon_1^2$$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 3

Shrinking Lemma $\Rightarrow \exists U_1 \in \mathcal{G}^{\ell_1}$ s.t.

$$||\Delta_1 - U_1|| = \left|\left|UU_0^\dagger - U_1\right|\right| = ||U - U_1U_0|| < \varepsilon_1^2$$

Define $\Delta_2 := \Delta_1 U_1^\dagger = UU_0^\dagger U_1^\dagger$. Then:

$$||\Delta_2 - I|| = \left|\left|(U - U_1U_0)U_0^\dagger U_1^\dagger\right|\right| = ||U - U_1U_0|| < \varepsilon_1^2 < \varepsilon_2$$

$\Rightarrow \Delta_2 \in S_{\varepsilon_2}$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 3

Shrinking Lemma $\Rightarrow \exists U_1 \in \mathcal{G}^{\ell_1}$ s.t.

$$||\Delta_1 - U_1|| = \left|\left|UU_0^\dagger - U_1\right|\right| = ||U - U_1 U_0|| < \varepsilon_1^2$$

Define $\Delta_2 := \Delta_1 U_1^\dagger = UU_0^\dagger U_1^\dagger$. Then:

$$||\Delta_2 - I|| = \left|\left|(U - U_1 U_0)U_0^\dagger U_1^\dagger\right|\right| = ||U - U_1 U_0|| < \varepsilon_1^2 < \varepsilon_2$$

$\Rightarrow \Delta_2 \in S_{\varepsilon_2} \quad \ldots$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 3

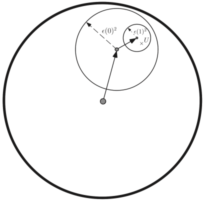
Shrinking Lemma $\Rightarrow \exists U_1 \in \mathcal{G}^{\ell_1}$ s.t.

$$||\Delta_1 - U_1|| = \left|\left|UU_0^\dagger - U_1\right|\right| = ||U - U_1U_0|| < \varepsilon_1^2$$

Define $\Delta_2 := \Delta_1 U_1^\dagger = UU_0^\dagger U_1^\dagger$. Then:

$$||\Delta_2 - I|| = \left|\left|(U - U_1U_0)U_0^\dagger U_1^\dagger\right|\right| = ||U - U_1U_0|| < \varepsilon_1^2 < \varepsilon_2$$

$\Rightarrow \Delta_2 \in S_{\varepsilon_2} \quad \ldots$



1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 4

After k steps: $U_k \in \mathcal{G}^{\ell_k}$ s.t. $||U - U_k U_{k-1} \dots U_0|| < \varepsilon_k^2$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 4

After k steps: $U_k \in \mathcal{G}^{\ell_k}$ s.t. $||U - U_k U_{k-1} \dots U_0|| < \varepsilon_k^2$
$\#(\text{gates}) = \sum_{m=0}^{k} \ell_m = \sum_{m=0}^{k} 5^m \ell_0 = \frac{5^{k+1}-1}{4}\ell_0 < \frac{5}{4}5^k \ell_0$ with accuracy ε_k^2.
What is k?

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 4

After k steps: $U_k \in \mathcal{G}^{\ell_k}$ s.t. $||U - U_k U_{k-1} \dots U_0|| < \varepsilon_k^2$
$\#(\text{gates}) = \sum_{m=0}^{k} \ell_m = \sum_{m=0}^{k} 5^m \ell_0 = \frac{5^{k+1}-1}{4} \ell_0 < \frac{5}{4} 5^k \ell_0$ with accuracy ε_k^2.
What is k?

$$\varepsilon_k^2 = \left((s\varepsilon_0)^{(3/2)^k} / s \right)^2 = \varepsilon$$

Solve for k:

$$\left(\frac{3}{2} \right)^k = \frac{\log(1/s^2 \varepsilon)}{2 \log(1/s\varepsilon_0)} = 5^{k/c}$$

for $c \approx 4$.

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Proof Solovay-Kitaev: Step 4

After k steps: $U_k \in \mathcal{G}^{\ell_k}$ s.t. $||U - U_k U_{k-1} \dots U_0|| < \varepsilon_k^2$
$\#(\text{gates}) = \sum_{m=0}^{k} \ell_m = \sum_{m=0}^{k} 5^m \ell_0 = \frac{5^{k+1}-1}{4} \ell_0 < \frac{5}{4} 5^k \ell_0$ with accuracy ε_k^2.
What is k?

$$\varepsilon_k^2 = \left((s\varepsilon_0)^{(3/2)^k} / s \right)^2 = \varepsilon$$

Solve for k:

$$\left(\frac{3}{2}\right)^k = \frac{\log(1/s^2\varepsilon)}{2\log(1/s\varepsilon_0)} = 5^{k/c}$$

for $c \approx 4$.

$$\#(\text{gates}) < \frac{5}{4} 5^k \ell_0 = \frac{5}{4} \left(\frac{3}{2}\right)^{kc} \ell_0 = \frac{5}{4} \left(\frac{\log(1/s^2\varepsilon)}{2\log(1/s\varepsilon_0)} \right)^c \ell_0 = \mathcal{O}(\log^c(1/\varepsilon))$$

1. Choose ε_0 wisely
2. $\langle \mathcal{G} \rangle$ dense in $\mathrm{SU}(2) \Rightarrow$ Choose ℓ_0 s.t. $\mathcal{G}^{\ell_0}$ is ε_0^2-net for $\mathrm{SU}(2)$.
3. Apply Shrinking Lemma repeatedly
4. Stop if given accuracy is reached

Shrinking Lemma

Lemma

There exist ε', s *s.t. for any* $\mathcal{G}$ *and* $\varepsilon \leq \varepsilon'$ *we have: If* $\mathcal{G}^{\ell}$ *is an* ε^2*-net for* S_ε *then* $\mathcal{G}^{5\ell}$ *is an* $s\varepsilon^3$*-net for* $S_{\sqrt{s\varepsilon^3}}$

Shrinking Lemma

Lemma

There exist ε', s s.t. for any $\mathcal{G}$ and $\varepsilon \leq \varepsilon'$ we have: If $\mathcal{G}^{\ell}$ is an ε^2-net for S_ε then $\mathcal{G}^{5\ell}$ is an $s\varepsilon^3$-net for $S_{\sqrt{s\varepsilon^3}}$

To prove this lemma, we have to transform the parameters $(\ell, \varepsilon^2, \varepsilon) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Goal: Approximate U in S_{ε^2}

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Goal: Approximate U in S_{ε^2}

Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Goal: Approximate U in S_{ε^2}

Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$

Problem: complicated operation

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Goal: Approximate U in S_{ε^2}

Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$

Problem: complicated operation

Fact: Near identity we can use matrix commutator $[A, B] = AB - BA$ instead of group commutator

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate U in S_{ε^2}
Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$
Problem: complicated operation
Fact: Near identity we can use matrix commutator $[A, B] = AB - BA$ instead of group commutator

$$\begin{tikzcd} V = e^{-iA}, W = e^{-iB} \arrow[r, "{[\![\cdot,\cdot]\!]}"] & {[\![V, W]\!]} \\ A, B \arrow[u] \arrow[r, "{[\cdot,\cdot]}"] & {[A, B]} \arrow[u, dotted, "?"] \end{tikzcd}$$

$$||A|| < \varepsilon, ||B|| < \varepsilon, \left|\left| [\![e^{-iA}, e^{-iB}]\!] - e^{-[A,B]} \right|\right| \leq \mathcal{O}(\varepsilon^3)$$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$

Goal: Approximate U in S_{ε^2}

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate U in S_{ε^2}
Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$
Matrix commutator for $\mathrm{SU}(2)$: $V = u(\vec{a}) := e^{-\frac{i}{2}\vec{a}\cdot\vec{\sigma}}, W = u(\vec{b}) = e^{-\frac{i}{2}\vec{b}\cdot\vec{\sigma}}$ where
$\vec{r}\cdot\vec{\sigma} = r_x X + r_y Y + r_z Z$

$$[X, Y] = 2iZ, [Y, Z] = 2iX, [Z, X] = 2iY \Rightarrow [\vec{a}\cdot\vec{\sigma}, \vec{b}\cdot\vec{\sigma}] = 2i(\vec{a}\times\vec{b})\vec{\sigma}$$

$$u(\vec{a}\times\vec{b}) = e^{-\left[\frac{1}{2}\vec{a}\cdot\vec{\sigma}, \frac{1}{2}\vec{b}\cdot\vec{\sigma}\right]}$$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate U in S_{ε^2}
Idea: Use Group commutator $[\![V, W]\!] = VWV^\dagger W^\dagger$
Matrix commutator for SU(2): $V = u(\vec{a}) := e^{-\frac{i}{2}\vec{a}\cdot\vec{\sigma}}, W = u(\vec{b}) = e^{-\frac{i}{2}\vec{b}\cdot\vec{\sigma}}$ where
$\vec{r} \cdot \vec{\sigma} = r_x X + r_y Y + r_z Z$

$$[X, Y] = 2iZ, [Y, Z] = 2iX, [Z, X] = 2iY \Rightarrow [\vec{a} \cdot \vec{\sigma}, \vec{b} \cdot \vec{\sigma}] = 2i(\vec{a} \times \vec{b})\vec{\sigma}$$

$$u(\vec{a} \times \vec{b}) = e^{-\left[\frac{1}{2}\vec{a}\cdot\vec{\sigma}, \frac{1}{2}\vec{b}\cdot\vec{\sigma}\right]}$$

$$\Rightarrow \left|\left|[\![V, W]\!] - u(\vec{a} \times \vec{b})\right|\right| = \mathcal{O}(\varepsilon^3)$$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate $U = u(\vec{x})$ in $S_{\varepsilon^2}, |\vec{x}| < \varepsilon^2$
Main Idea:

- Write $\vec{x} = \vec{y} \times \vec{z}$ with $|\vec{y}|, |\vec{z}| \leq \varepsilon$
- Approximate $u(\vec{y}), u(\vec{z})$ with $\vec{y_0}, \vec{z_0}$ s.t. $u(\vec{y_0}), u(\vec{z_0}) \in \mathcal{G}^\ell$ is ε^2-approximation

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate $U = u(\vec{x})$ in $S_{\varepsilon^2}, |\vec{x}| < \varepsilon^2$
Main Idea:

- Write $\vec{x} = \vec{y} \times \vec{z}$ with $|\vec{y}|, |\vec{z}| \leq \varepsilon$
- Approximate $u(\vec{y}), u(\vec{z})$ with $\vec{y_0}, \vec{z_0}$ s.t. $u(\vec{y_0}), u(\vec{z_0}) \in \mathcal{G}^\ell$ is ε^2-approximation

$$||u(\vec{x}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq ||u(\vec{x}) - u(\vec{y_0} \times \vec{z_0})|| + ||u(\vec{y_0} \times \vec{z_0}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq s\varepsilon^3$$

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate $U = u(\vec{x})$ in $S_{\varepsilon^2}, |\vec{x}| < \varepsilon^2$
Main Idea:

- Write $\vec{x} = \vec{y} \times \vec{z}$ with $|\vec{y}|, |\vec{z}| \leq \varepsilon$
- Approximate $u(\vec{y}), u(\vec{z})$ with $\vec{y_0}, \vec{z_0}$ s.t. $u(\vec{y_0}), u(\vec{z_0}) \in \mathcal{G}^\ell$ is ε^2-approximation

$$||u(\vec{x}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq ||u(\vec{x}) - u(\vec{y_0} \times \vec{z_0})|| + ||u(\vec{y_0} \times \vec{z_0}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq s\varepsilon^3$$

$\Rightarrow [\![u(\vec{y_0}), u(\vec{z_0})]\!]$ $s\varepsilon^3$-approximates U in 4ℓ gates $\Rightarrow s\varepsilon^3$-net for S_{ε^2}

Proof Shrinking Lemma

$(\ell, \varepsilon^2, \varepsilon) \mapsto (4\ell, s\varepsilon^3, \varepsilon^2) \mapsto (5\ell, s\varepsilon^3, \sqrt{s\varepsilon^3})$
Goal: Approximate $U = u(\vec{x})$ in $S_{\varepsilon^2}, |\vec{x}| < \varepsilon^2$
Main Idea:

- Write $\vec{x} = \vec{y} \times \vec{z}$ with $|\vec{y}|, |\vec{z}| \leq \varepsilon$
- Approximate $u(\vec{y}), u(\vec{z})$ with $\vec{y_0}, \vec{z_0}$ s.t. $u(\vec{y_0}), u(\vec{z_0}) \in \mathcal{G}^\ell$ is ε^2-approximation

$$||u(\vec{x}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq ||u(\vec{x}) - u(\vec{y_0} \times \vec{z_0})|| + ||u(\vec{y_0} \times \vec{z_0}) - [\![u(\vec{y_0}), u(\vec{z_0})]\!]|| \leq s\varepsilon^3$$

$\Rightarrow [\![u(\vec{y_0}), u(\vec{z_0})]\!]$ $s\varepsilon^3$-approximates U in 4ℓ gates $\Rightarrow s\varepsilon^3$-net for S_{ε^2}
Now: Perform translation step: Given $U \in S_{\sqrt{s\varepsilon^3}}$ we can find $V \in \mathcal{G}^\ell$ s.t.
$||U - V|| \leq \varepsilon^2 \Rightarrow UV^\dagger \in S_{\varepsilon^2}$
Find $W_1, W_2 \in \mathcal{G}^\ell$ s.t. $||[\![W_1, W_2]\!] - UV^\dagger|| \leq s\varepsilon^3 \Rightarrow ||[\![W_1, W_2]\!]V - U|| \leq s\varepsilon^3$

Table of Contents

Idea inverse-free Solovay-Kitaev

Original Solovay-Kitaev: We only have ε-approximations to unitaries (from previous recursive step). We can multiply them. Gate set needs to be inverse-closed.
Goal: Find correct sequence to get higher precision.

Idea inverse-free Solovay-Kitaev

Original Solovay-Kitaev: We only have ε-approximations to unitaries (from previous recursive step). We can multiply them. Gate set needs to be inverse-closed.
Goal: Find correct sequence to get higher precision.
Now: Solovay-Kitaev without inverses
2016: Sardharwalla, Cubitt, Harrow, Linden: Approximate inverses with $\mathcal{O}(\varepsilon^2)$-precision suffices. Pauli group can be used.

Idea inverse-free Solovay-Kitaev

Original Solovay-Kitaev: We only have ε-approximations to unitaries (from previous recursive step). We can multiply them. Gate set needs to be inverse-closed.
Goal: Find correct sequence to get higher precision.
Now: Solovay-Kitaev without inverses
2016: Sardharwalla, Cubitt, Harrow, Linden: Approximate inverses with $\mathcal{O}(\varepsilon^2)$-precision suffices. Pauli group can be used.
How to do in general?

Self-correcting sequences

Definition

Consider operators $\{g_1, \ldots, g_k\} \subset \mathrm{SU}(d)$ and set of corresponding ε-approximate operators $\{g'_1, \ldots, g'_k\} \subset \mathrm{SU}(d)$ s.t. $||g'_i - g_i|| \leq \varepsilon$. A self-correcting sequence is a word in the approximate operators which approximate the identity to a higher order in ε

$$g'_{i_1} \ldots g'_{i_N} = I + \mathcal{O}(\varepsilon^n) \quad n > 1$$

Bouland, Giurgica-Tiron (2021): There exists quadratically-precise sequence in $\mathrm{SU}(d)$

Bouland, Giurgica-Tiron

Use Pauli approximations

$$X' = X + \mathcal{O}(\varepsilon)$$
$$Z' = Z + \mathcal{O}(\varepsilon)$$

Dimension $d = 2$: $Z'X'X'Z'X'Z'Z'X' = I + \mathcal{O}(\varepsilon^2) \quad N = 8$

Bouland, Giurgica-Tiron

Use Pauli approximations

$$X' = X + \mathcal{O}(\varepsilon)$$
$$Z' = Z + \mathcal{O}(\varepsilon)$$

Dimension $d = 2$: $Z'X'X'Z'X'Z'Z'X' = I + \mathcal{O}(\varepsilon^2) \quad N = 8$

Dimension $d \geq 2$: $\left(Z'X'^d\right)^{d-1} Z' \left(X'Z'^d\right)^{d-1} X' = I + \mathcal{O}(\varepsilon^2) \quad N = 2d^2$

How to invert U: We have $X' = X + \mathcal{O}(\varepsilon), Z' = Z + \mathcal{O}(\varepsilon), \hat{U}^\dagger = U^\dagger + \mathcal{O}(\varepsilon)$

Bouland, Giurgica-Tiron

Use Pauli approximations

$$X' = X + \mathcal{O}(\varepsilon)$$
$$Z' = Z + \mathcal{O}(\varepsilon)$$

Dimension $d = 2$: $Z'X'X'Z'X'Z'Z'X' = I + \mathcal{O}(\varepsilon^2) \quad N = 8$

Dimension $d \geq 2$: $\left(Z'X'^d\right)^{d-1} Z' \left(X'Z'^d\right)^{d-1} X' = I + \mathcal{O}(\varepsilon^2) \quad N = 2d^2$

How to invert U: We have $X' = X + \mathcal{O}(\varepsilon), Z' = Z + \mathcal{O}(\varepsilon), \hat{U}^\dagger = U^\dagger + \mathcal{O}(\varepsilon)$

$X'\hat{U}^\dagger U = X + \mathcal{O}(\varepsilon)$

Bouland, Giurgica-Tiron

Use Pauli approximations

$$X' = X + \mathcal{O}(\varepsilon)$$
$$Z' = Z + \mathcal{O}(\varepsilon)$$

Dimension $d = 2$: $Z'X'X'Z'X'Z'Z'X' = I + \mathcal{O}(\varepsilon^2) \quad N = 8$

Dimension $d \geq 2$: $\left(Z'X'^d\right)^{d-1} Z' \left(X'Z'^d\right)^{d-1} X' = I + \mathcal{O}(\varepsilon^2) \quad N = 2d^2$

How to invert U: We have $X' = X + \mathcal{O}(\varepsilon), Z' = Z + \mathcal{O}(\varepsilon), \hat{U}^\dagger = U^\dagger + \mathcal{O}(\varepsilon)$

$X'\hat{U}^\dagger U = X + \mathcal{O}(\varepsilon)$

Let $J(X', Z') = I + \mathcal{O}(\varepsilon^2)$ be a self-correcting sequence in X', Z'.

$\Rightarrow J(X'\hat{U}^\dagger U, Z') = I + \mathcal{O}(\varepsilon^2)$

Bouland, Giurgica-Tiron

Use Pauli approximations

$$X' = X + \mathcal{O}(\varepsilon)$$
$$Z' = Z + \mathcal{O}(\varepsilon)$$

Dimension $d = 2$: $Z'X'X'Z'X'Z'Z'X' = I + \mathcal{O}(\varepsilon^2) \quad N = 8$

Dimension $d \geq 2$: $\left(Z'X'^d\right)^{d-1} Z' \left(X'Z'^d\right)^{d-1} X' = I + \mathcal{O}(\varepsilon^2) \quad N = 2d^2$

How to invert U: We have $X' = X + \mathcal{O}(\varepsilon), Z' = Z + \mathcal{O}(\varepsilon), \hat{U}^\dagger = U^\dagger + \mathcal{O}(\varepsilon)$

$X'\hat{U}^\dagger U = X + \mathcal{O}(\varepsilon)$

Let $J(X', Z') = I + \mathcal{O}(\varepsilon^2)$ be a self-correcting sequence in X', Z'.

$\Rightarrow J(X'\hat{U}^\dagger U, Z') = I + \mathcal{O}(\varepsilon^2)$

This sequence is close to identity and in an instance of U itself. Remove U and done ;)

Consequences & Open problems

Consequences:

- Sequence for inverses has length $\mathcal{O}(d^2)$
  $\Rightarrow \#(\text{gates}) = \mathcal{O}(\log^c(1/\varepsilon)), c = \mathcal{O}(\log d)$.
- simplifies proofs in various areas of quantum complexity theory
- Construction could be practically useful when errors are coherent e.g. in dynamic decoupling

Open problems:

- Reduce exponent from $\mathcal{O}(\log d)$ to the nonconstructive upper bound of 3.
- Understand mathematics of self-correcting sequences. Generalize to other groups and higher orders
- ...

Solovay-Kitaev Rap by ChatGPT

Yo, let me tell you about a theorem so neat
It's called the Solovay-Kitaev, let's take a seat
It's about quantum gates and approximation, you see
Making quantum computing even better, that's the key

Solovay-Kitaev, Solovay-Kitaev
Universal quantum gates, we can achieve
Polynomial complexity, that's the key
Approximation with precision, can't you see?

For any finite group G and positive ϵ
We can approximate any U-gate with precision
Using a finite set of quantum gates, we can't go wrong
Polylogarithmic complexity, won't take too long

Solovay-Kitaev, Solovay-Kitaev
Universal quantum gates, we can achieve
Polynomial complexity, that's the key
Approximation with precision, can't you see?

With Solovay-Kitaev, we can compute with ease
More complex operations, our limits will increase
Like a puzzle, we fit the gates to get the right solution
And quantum computing will become a revolution

Solovay-Kitaev, Solovay-Kitaev
Universal quantum gates, we can achieve
Polynomial complexity, that's the key
Approximation with precision, can't you see?

So let's celebrate Solovay-Kitaev, let's give it a cheer
For the future of quantum computing is looking so clear!

Thank You! Any Questions?