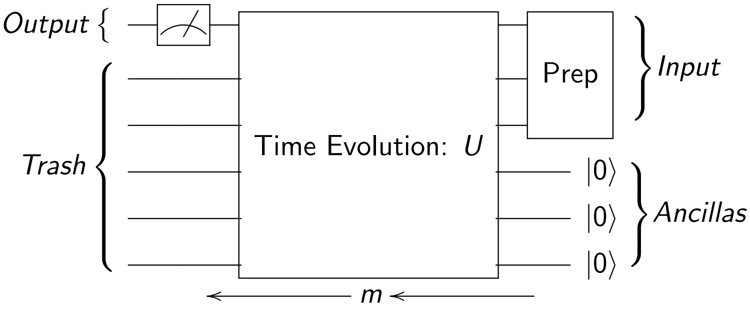
"Improved classical simulation of quantum circuits dominated by Clifford gates"

Presented by Patrick Rall

Publication by Sergey Bravyi and David Gosset on Jan 29, 2016

October 14, 2016

Model of Computation



- System: input qubits + ancilla qubits = n qubits
- Hilbert space: $\mathbb{C}^{2n}$ - exponentially large
- Intuition: Quantum computation efficiently calculates matrix multiplication $U|\Psi_{in}\rangle$

Model of Computation: Output Probabilities

- Simplification: assume input state $\left|0^{\otimes n}\right\rangle$
- Measure output qubit: some probability distribution
- Goal: sample from this distribution

$$c_x = \left\langle x \middle| U \middle| 0^{\otimes n} \right\rangle$$

$$P_x = |c_x|^2 = \left\langle 0^{\otimes n} \middle| U^\dagger |x\rangle\langle x| U \middle| 0^{\otimes n} \right\rangle$$

- Here $|x\rangle\langle x|$ is a projector onto an output state

Classical simulation: naïve approach

- Given $U = U_m \ldots U_3 U_2 U_1$, calculate P_x:

$$P_x = \langle 0^{\otimes n} \big| U_1^\dagger U_2^\dagger \ldots U_m^\dagger | x \rangle \langle x | U_m \ldots U_2 U_1 \big| 0^{\otimes n} \rangle$$

- Calculate m matrix multiplications in $\mathbb{C}^{2n}$
- Naïve runtime: $m(2^n)^\omega$, best known[1] $\omega = 2.3737$

Interpretation

- Exponential in n: always intractable for large enough n
- Getting rid of exponentiality? Would imply:
  Quantum computing = Classical computing
- Algorithm 'moves' exponent in n to other parameter

[1] Coppersmith-Winograd algorithm

Stabilizer States

- D. Gottesman (1998): "The Heisenberg Representation of Quantum Computers"[2]
- Consider an Abelian subgroup $G \subset \mathcal{P}_n$ with $-I \notin G$.
- Def: $|\phi\rangle$ is **stabilized** by G if $P|\phi\rangle = |\phi\rangle, \forall P \in G$.
- Def: $|\phi\rangle$ is a **stabilizer state** if stabilized by some G
- Clifford gates map stabilizer states to stabilizer states

- Example: $G = \langle X \otimes X, Z \otimes Z \rangle \subset \mathcal{P}_2$
- Unique stabilizer state:

$$|\phi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- Degrees of freedom: G defined by k **stabilizer generators** stabilizes 2^{n-k} states.

[2] https://arxiv.org/abs/quant-ph/9807006v1

The Gottesmann-Knill Theorem

- Stabilizer projector: project onto space stabilized by G

$$\Pi_G = \prod_{P \in G} \frac{1+P}{2}$$

- Clifford gates can act on stabilizer projectors: act on each generator of G
- Want to calculate:

$$P_x = \left\langle 0^{\otimes n} \middle| U_1^\dagger U_2^\dagger \ldots U_m^\dagger |x\rangle\langle x| U_m \ldots U_2 U_1 \middle| 0^{\otimes n} \right\rangle$$

- What if $U_i \in \mathcal{C}_n$? Then, given $|x\rangle\langle x| = \Pi_x$:

$$P_x = \left\langle 0^{\otimes n} \middle| U_1^\dagger U_2^\dagger \ldots \Pi_{G_m(x)} \ldots U_2 U_1 \middle| 0^{\otimes n} \right\rangle$$

$$= \left\langle 0^{\otimes n} \middle| U_1^\dagger \Pi_{G_2(x)} U_1 \middle| 0^{\otimes n} \right\rangle = \left\langle 0^{\otimes n} \middle| \Pi_{G(x)} \middle| 0^{\otimes n} \right\rangle$$

- Result: can calculate circuit in polynomial time!

Clifford + T: A universal gate set

- Circuits composed of Cliffords, i.e. $H, S, CNOT$, can be simulated efficiently
- $\{H, S, CNOT\}$ **acting on stabilizer states only** is not universal
  - Mathematical fact
  - Otherwise: quantum computing = classical computing
- Add the T gate to obtain universal gate set:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

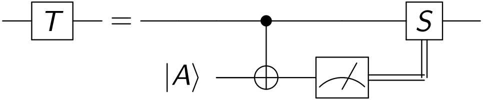
- Problem: T gate hard to simulate classically
- Incidentally: T gate also hard to build in experiment

Gadgetization with Magic States

- Consider a ‘magic state’:

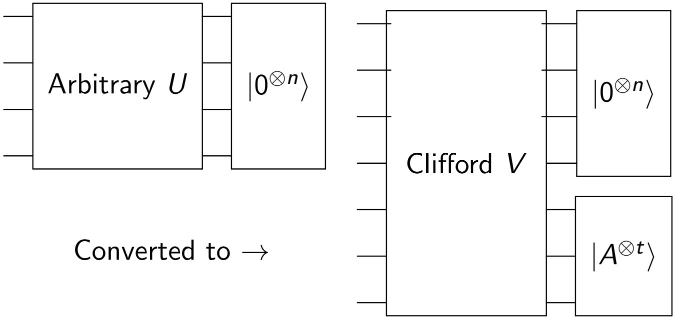
$$|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$$

- Use $|A\rangle$ as a resource to write T in terms of Cliffords:



- Measurement destroys magic state in the process.
- Input to circuit was $|0^{\otimes n}\rangle$, now is $|0^{\otimes n}A^{\otimes t}\rangle$.
- t = number of T gates in circuit

Converting the problem



The challenge

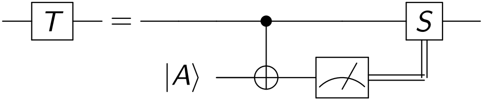
- Before: non-Clifford circuit with T gates
- After: non-stabilizer 'magic' resource state $|A^{\otimes t}\rangle$

Algorithm Goal

- Goal: Sample from probability distribution

$$P_x = \langle 0^{\otimes n} | U^\dagger \Pi_X U | 0^{\otimes n} \rangle$$

- Gadgetize non-Clifford unitary U to Clifford V with 'magic' resource state $|A^{\otimes t}\rangle$



- How to deal with measurement? Post-select measurement outcomes into string y. Calculate:

$$P_x = \frac{\langle A^{\otimes t} 0^{\otimes n} | V^\dagger (\Pi_x \otimes \Pi_y) V | 0^{\otimes n} A^{\otimes t} \rangle}{\langle A^{\otimes t} 0^{\otimes n} | V^\dagger (\mathbb{I} \otimes \Pi_y) V | 0^{\otimes n} A^{\otimes t} \rangle}$$

- Works for any y!

Concept: Stabilizer Rank χ

- Remaining problem: non-stabilizer state $|A^{\otimes t}\rangle$
- Write as a linear combination of χ stabilizer states $|\phi_a\rangle$

$$\left|A^{\otimes t}\right\rangle \approx \sum_{a}^{\chi} z_a |\phi_a\rangle = |\Psi\rangle$$

- 2^t stabilizer states: Naïve upper bound $\chi \leq 2^t$
- Clever trick 1: Recognize that $\left|A^{\otimes 2}\right\rangle$ is a sum of two stabilizer states. Divide $|A^{\otimes t}\rangle$ into pairs: $\chi \leq 2^{t/2}$
- Clever trick 2 (see appendix): Achieve $\chi \sim O(2^{0.23t})$. Authors conjecture that this is optimal.

Summary Calculation

$$P_x = \left\langle 0^{\otimes n} \middle| U^\dagger \Pi_x U \middle| 0^{\otimes n} \right\rangle$$

$$= \frac{\langle A^{\otimes t} 0^{\otimes n} | V^\dagger (\Pi_x \otimes \Pi_y) V | 0^{\otimes n} A^{\otimes t} \rangle}{\langle A^{\otimes t} 0^{\otimes n} | V^\dagger (\mathbb{I} \otimes \Pi_y) V | 0^{\otimes n} A^{\otimes t} \rangle}$$

$$= \frac{\langle A^{\otimes t} 0^{\otimes n} | \Pi_{G(x,y)} | 0^{\otimes n} A^{\otimes t} \rangle}{\langle A^{\otimes t} 0^{\otimes n} | \Pi_{H(y)} | 0^{\otimes n} A^{\otimes t} \rangle} = \frac{1}{2^u} \frac{\langle A^{\otimes t} | \Pi_{\bar{G}(x,y)} | A^{\otimes t} \rangle}{\langle \bar{A}^{\otimes t} \middle| \Pi_{H(y)} \middle| A^{\otimes t} \rangle}$$

$$= \frac{1}{2^u} \frac{|\Pi_{\bar{G}(x,y)} | A^{\otimes t} \rangle|^2}{|\Pi_{\bar{H}(y)} | A^{\otimes t} \rangle|^2} \approx \frac{1}{2^u} \frac{|\Pi_{\bar{G}(x,y)} | \Psi \rangle|^2}{|\Pi_{\bar{H}(y)} | \Psi \rangle|^2}$$

- Calculation boils down to $|\Pi_{\bar{G}(x,y)}|\Psi\rangle|^2$ and $|\Pi_{\bar{H}(y)}|\Psi\rangle|^2$
- Approx. requires random y, rather than arbitrary y

The Algorithm

1. Choose random y, evaluate projectors $\Pi_{\bar{G}(x,y)}$, $\Pi_{\bar{H}(y)}$
2. Compute $|A^{\otimes t}\rangle \approx \sum_a^{\chi} z_a|\phi_a\rangle = |\Psi\rangle$
   such that $|\langle A^{\otimes t}|\Psi\rangle|^2 \geq 1-\delta$,
   where $|\phi_a\rangle$ are stabilizer states, $\chi = O(2^{0.23t}\delta^{-1})$
3. Evaluate inner products $\left|\Pi_{\bar{G}(x,y)}|\Psi\rangle\right|^2$ and $\left|\Pi_{\bar{H}(y)}|\Psi\rangle\right|^2$

$$|\Pi|\Psi\rangle|^2 = \left|\Pi\sum_a^{\chi} z_a|\phi_a\rangle\right|^2 = \left|\sum_a^{\chi} z_a\Pi|\phi_a\rangle\right|^2$$

4. Compute distribution $P_{x=0}, P_{x=1} = 1 - P_{x=0}$, and sample from distribution.

Runtime

- Sample from output distribution for a string x:

$$\text{poly}(n, m) + 2^{0.23t} t^3 w^4$$

  - Exponential: number of T gates t
  - Polynomial: n qubits, width m circuit
  - Length of output string $|x| = w$
- Projector Π_x has 2^w generators: w^4 via trick (appendix)
- Exponential part is highly parallelizable
  - Each term in $\sum_a^{\chi} z_a \Pi |\phi_a\rangle$ can be calculated independently

Conclusions, next steps

Implementation

- MATLAB implementation by Bravyi, Gosset
  - Hidden shift algorithm on a laptop
  - 40 qubits, 50 T gates
- Python+C implementation by Iskren Vankov, me
- Upcoming: CUDA implementation?

New concept: Stabilizer Rank

- How to decompose arbitrary $|\Psi\rangle$ into stabilizer states?
- Improve naïve runtime $O(m2^n)$ to $O(m2^{\alpha n})$ for $\alpha < 1$?

Appendix

- Alexei Kitaev's Stabilizer Toolkit
- Sampling larger bitstrings x
- Stabilizer decomposition of $|A^{\otimes t}\rangle$
- Computing inner products in $O(\chi)$ rather than $O(\chi^2)$

Alexei Kitaev's Stabilizer Toolkit

- Traditional representation: $G \subset \mathcal{P}_n$
- Efficient representation:[3]
  - Affine space $\mathcal{K}$: Subspace of $\mathbb{F}^2$ such that $\mathcal{L}(\mathcal{K}) = h \oplus \mathcal{K}$
  - Quadratic form q: Function $q : \mathcal{K} \to \mathbb{Z}_8$ with properties

$$|\mathcal{K}, q\rangle = 2^{-k/2} \sum_{x \in \mathcal{K}} e^{\frac{i\pi}{4} q(x)} |x\rangle$$

- Algorithms:
  - Inner product of two states in $O(n^3)$
  - Measure a Pauli operator in $O(n^2)$
  - Sample random stabilizer states in $O(n^2)$ on average ($O(n^3)$ worst case)

[3] H. J. Garca and I. L. Markov, "Hybrid Techniques for Simulating Quantum Circuits using the Heisenberg Representation"

Sampling larger bitstrings x

- Projector Π_x has 2^w generators. Contributes to $|\Pi|\Psi\rangle|^2$.
- Achieve polynomial time: Sample first bit of x, x_1, then evaluate conditional probability for next bit, etc:

$$x = x_1 x_2 \ldots x_w$$

$$P(x_2|x_1) = \frac{P(x_1, x_2)}{P(x_1)} \rightarrow P(x_3|x_1 x_2) = \frac{P(x_1, x_2, x_3)}{P(x_1, x_2)} \ldots$$

- w^4: not particularly fast, but at least not exponential

Stabilizer decomposition of $|A^{\otimes t}\rangle$

$$|A^{\otimes t}\rangle \approx \sum_a^{\chi} z_a|\phi_a\rangle = |\Psi\rangle$$

$$|\langle A^{\otimes t}|\Psi\rangle|^2 \geq 1-\delta$$

- Low stabilizer rank decomposition $\chi = O(2^{0.23t}\delta^{-1})$

$$|A\rangle = e^{i\pi/8}HS^\dagger\left(\cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle\right) = e^{i\pi/8}HS^\dagger|H\rangle$$

- Find $|\mathcal{L}\rangle$ such that $|\langle H^{\otimes t}|\mathcal{L}\rangle|^2 \geq 1-\delta$
- Choose a dimension k such that $4 \geq 2^k v^{2t}\delta \geq 2$, $v = \cos\frac{\pi}{8}, |H\rangle = \frac{1}{2v}(|0\rangle + |+\rangle)$
- Sample a random subspace $\mathcal{L} \subset \mathbb{F}_2^n$ with dimension k. From Markov's inequality:

$$\Pr\left[|\langle H^{\otimes t}|\mathcal{L}\rangle|^2 \geq 1-\delta\right] \geq \Omega(\delta)$$

- Keep sampling $\mathcal{L}$ until this is true. Should take $O(\delta^{-1})$.

Computing inner products in $O(\chi)$ rather than $O(\chi^2)$

- Given $\Pi|\Psi\rangle = \sum_a^\chi z_a \Pi|\phi_a\rangle$, compute $|\Pi|\Psi\rangle|^2$
- Naïve method: $O(\chi^2)$ calculations of $z_a z_b \langle\phi_a|\Pi|\phi_b\rangle$
- With clever trick: $O(\chi)$ calculations of $z_a\langle\theta_i|\Pi|\phi_a\rangle$ with $L = 1/p_f\epsilon$ random states $|\theta_i\rangle$

$$\text{Random variable } \alpha = \frac{2^t}{L}\sum_{i=1}^{L} |\langle\theta_i|\Pi|\Psi\rangle|^2$$

$$\Pr\left[(1-\epsilon)|\Pi|\Psi\rangle|^2 \leq \alpha \leq (1+\epsilon)|\Pi|\Psi\rangle|^2\right] \geq 1 - p_f$$

- Derivation uses that stabilizer states $\mathcal{S}$ are a 2-design

$$\sum_{\theta\in\mathcal{S}} (|\theta\rangle\langle\theta| \otimes |\theta\rangle\langle\theta|) = \int_{\text{Haar}} (|\phi\rangle\langle\phi| \otimes |\phi\rangle\langle\phi|) d\phi$$