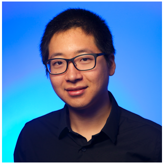
Toward the first quantum simulation with quantum speedup

Yuan Su
University of Maryland






Andrew Childs
University of Maryland

Dmitri Maslov
University of Maryland
National Science Foundation

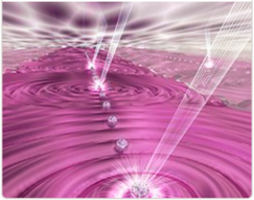
Yunseong Nam
University of Maryland
IonQ, Inc.

Neil J. Ross
University of Maryland
Dalhousie University

arXiv:1711.10980

Quantum advances

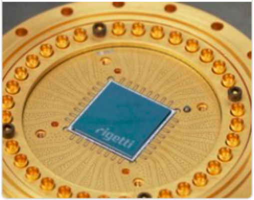
Quantum Simulators Wield Control Over More Than 50 Qubits, Setting New Record | ...



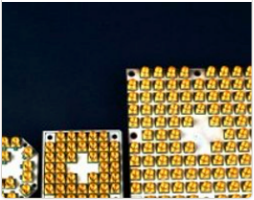
Microsoft bets on quantum computing to crack the world's toughest problems

IBM Raises the Bar with a 50-Qubit Quantum Computer

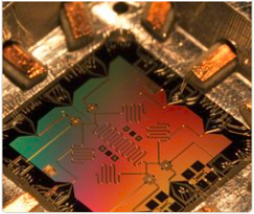
Rigetti has a 19 qubit quantum computing system and it runs unsupervised...



Intel Reveals Its New 49-Qubit Superconducting Quantum Chip at CES 2018



Revealed: Google's plan for quantum computer supremacy



The road to quantum computing

- Using a quantum computer to solve practical problems beyond the reach of classical computation may become possible in the foreseeable future.

- A near-term quantum computer may support:
  - tens of well-controlled qubits and
  - limited total number of gates that can be reliably performed.

- Therefore, reaching such a goal would require:
  - significant experimental advances and
  - **careful quantum algorithm design and implementation.**

Goals

Identify a problem that is

- practically relevant (not just quantum supremacy)
- classically intractable
- as easy as possible quantumly

Outline

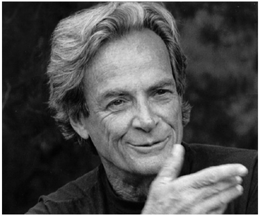
1. Quantum Simulation and Target System
2. Simulation Algorithms and New Techniques
3. Circuit Implementation and Results
4. Summary and Future Studies

Quantum simulation

Hamiltonian simulation problem

Given a description of the Hamiltonian H, an evolution time t, and an initial state $|\psi_0\rangle$, produce the final state $|\psi_t\rangle = e^{-iHt}|\psi_0\rangle$ up to some error ϵ.

- A quantum computer can prepare the final state efficiently if H is a local Hamiltonian.
- Upon measurement, it can efficiently answer questions that a classical one cannot.



"...nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

— Richard Feynman

What to simulate and why?

Heisenberg spin model on a ring

$H = \sum_{j=1}^{n}(\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j \sigma_j^z)$ with periodic boundary conditions and $h_j \in [-h, h]$ chosen uniformly at random.

- **Practicality:**
  - a model of self-thermalization and many-body localization
  - interesting among the condensed matter community
- **Classical intractability:**
  - thermalized/localized phase transition is poorly understood;
  - most extensive numerical study handled ≤ 25 spins.
- **Quantum tractability:**
  - could explore the transition by preparing a simple initial state, **evolving**, and performing a simple final measurement;
  - simulations of spin systems likely have low overhead.

System-size dependence

- For concreteness, choose $h_j \in [-1, 1]$, $t = n$, $\epsilon = 10^{-3}$ and $10 \leq n \leq 100$.

- Other choices of parameters may be possible, as long as the problem is still practically interesting and classically intractable, while remaining easy to solve quantumly.

- Our approach would apply to these alternative models essentially unchanged.

- With all parameters except n fixed, we study **the system-size dependence of quantum simulation algorithms.**

Complexity of simulation algorithms

- Recent algorithms have significantly improved asymptotic performance as a function of t and ϵ over the Trotter formula.
- We investigate **whether these recent algorithms are advantageous for simulating relatively small systems.**

Algorithm	Gate complexity (t, ϵ)	Gate complexity (n)
Product formula (PF), 1st order	$O(t^2/\epsilon)$	$O(n^5)$
Product formula (PF), $(2k)$th order	$O(5^{2k}t^{1+1/2k}/\epsilon^{1/2k})$	$O(5^{2k}n^{3+1/k})$
Quantum walk	$O(t/\sqrt{\epsilon})$	$O(n^4 \log n)$
Fractional-query simulation	$O\left(t\frac{\log^2(t/\epsilon)}{\log\log(t/\epsilon)}\right)$	$O\left(n^4\frac{\log n}{\log\log n}\right)$
Taylor series (TS)	$O\left(t\frac{\log^2(t/\epsilon)}{\log\log(t/\epsilon)}\right)$	$O\left(n^3\frac{\log^2 n}{\log\log n}\right)$
Linear combination of quantum walk	$O\left(t\frac{\log^{3.5}(t/\epsilon)}{\log\log(t/\epsilon)}\right)$	$O\left(n^4\frac{\log n}{\log\log n}\right)$
Quantum signal processing (QSP)	$O(t+\log(1/\epsilon))$	$O(n^3 \log n)$

Product formula algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \leq \alpha_\ell \leq 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- Can use the first-order PF:

$$\left\| e^{-it \sum_{j=1}^{L} \alpha_j H_j} - \left[\prod_{j=1}^{L} e^{-i \frac{t}{r} \alpha_j H_j} \right]^r \right\| \leq \frac{(Lt)^2}{r} \exp\left(\frac{L|t|}{r} \right)$$

- Generalizations to $(2k)$th order are known [Suzuki 92].
- The main challenge: choose explicit r such that error $\leq \epsilon$.

Product formula algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \le \alpha_\ell \le 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- Can use the first-order PF:

$$\left\| e^{-it\sum_{j=1}^{L}\alpha_j H_j} - \left[\prod_{j=1}^{L} e^{-i\frac{t}{r}\alpha_j H_j}\right]^r \right\| \le \frac{(Lt)^2}{r}\exp\left(\frac{L|t|}{r}\right)$$

- Generalizations to $(2k)$th order are known [Suzuki 92].
- The main challenge: choose explicit r such that error $\le \epsilon$.
- Analytic bound:

$$r_1 = \left\lceil \max\left\{Lt, \frac{e(Lt)^2}{\epsilon}\right\}\right\rceil$$

- Minimized bound:

$$r_1 = \min\left\{r : \frac{(Lt)^2}{r}\exp\left(\frac{Lt}{r}\right) \le \epsilon\right\}$$

- These bounds use the triangle inequality in a naive way.
- Is it possible to tighten the error analysis of PF?

Commutator bound

- Improve error analysis by **exploiting commutation relations.**

All terms commute
error $= 0$

Commutator bound

No term commutes
error $\leq \frac{(Lt)^2}{r}\exp(\frac{Lt}{r})$

- For $(2k)$th order PF, the commutator bound improves the n-dependence from $O(n^{3+1/k})$ to $O(n^{3+2/(2k+1)})$.

- Naive evaluation of the bound takes time $O(n^{2k+1})$.

- We further develop techniques that exploit the combinatorial structure of the Hamiltonian to compute the commutator bound in closed form.

Taylor series algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \leq \alpha_\ell \leq 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- Truncate the Taylor series:

$$e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} \approx \sum_{k=0}^{K} \frac{(-iHt)^k}{k!}$$

$$= \sum_{k=0}^{K} \sum_{\ell_1,\ldots,\ell_k=1}^{L} \frac{t^k}{k!} \alpha_{\ell_1} \cdots \alpha_{\ell_k} (-i)^k H_{\ell_1} \cdots H_{\ell_k}$$

$$= \sum_{j=0}^{\Gamma-1} \beta_j V_j$$

to get a linear combination of unitaries.

Taylor series algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \leq \alpha_\ell \leq 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- Truncate the Taylor series:

$$e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} \approx \sum_{k=0}^{K} \frac{(-iHt)^k}{k!}$$

$$= \sum_{k=0}^{K} \sum_{\ell_1,\ldots,\ell_k=1}^{L} \frac{t^k}{k!} \alpha_{\ell_1} \cdots \alpha_{\ell_k} (-i)^k H_{\ell_1} \cdots H_{\ell_k}$$

$$= \sum_{j=0}^{\Gamma-1} \beta_j V_j$$

to get a linear combination of unitaries.

- LCU [Berry et al., 14 & 15]: let

$$B|0\rangle = \frac{1}{\sqrt{s}} \sum_{j=0}^{\Gamma-1} \sqrt{\beta_j} |j\rangle$$

$$\text{select}(V) = \sum_{j=0}^{\Gamma-1} |j\rangle\langle j| \otimes V_j$$

then

$$(\langle 0|B^\dagger \otimes I)\text{select}(V)(B|0\rangle \otimes I) = \frac{1}{s} \sum_{j=0}^{\Gamma-1} \beta_j V_j$$

- OAA: alternate reflections along two subspaces to boost the scaled-down factor $\frac{1}{s}$.

select(V) synthesis

- The main challenge to synthesize $\sum_{j=0}^{\Gamma-1} |j\rangle\langle j| \otimes V_j$: generating all Boolean strings of length $\lceil \log_2 \Gamma \rceil$.
- Naive implementation requires $O(\Gamma \log \Gamma)$ gates.
- New idea: **walking on a binary tree**

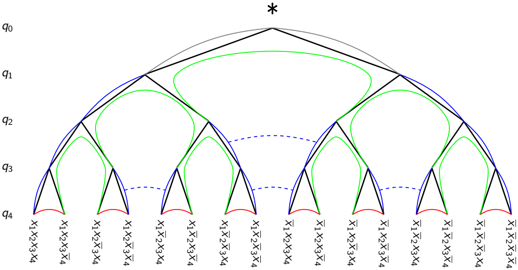
Example: generating 4-bit strings

$$x_1x_2x_3x_4$$

$$x_1x_2x_3\overline{x}_4$$

$$\vdots$$

$$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$$



- The new approach improves the gate complexity to $O(\Gamma)$, meeting a previously-established lower bound [Maslov 16].

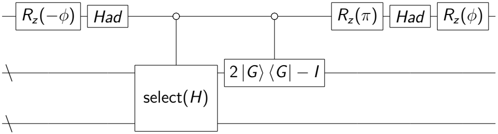
Quantum signal processing algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \leq \alpha_\ell \leq 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- "Encode" H into

$$|G\rangle = \frac{1}{\sqrt{\alpha}} \sum_{\ell=1}^{L} \sqrt{\alpha_\ell} |\ell\rangle$$

$$\mathrm{select}(H) = \sum_{\ell=1}^{L} |\ell\rangle\langle\ell| \otimes H_\ell$$

and construct V_ϕ as



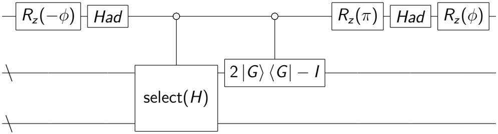
Quantum signal processing algorithm

- To simulate $H = \sum_{\ell=1}^{L} \alpha_\ell H_\ell$:
  - $0 \leq \alpha_\ell \leq 1$
  - H_ℓ is a tensor product of Paulis (up to a sign)
- "Encode" H into

$$|G\rangle = \frac{1}{\sqrt{\alpha}} \sum_{\ell=1}^{L} \sqrt{\alpha_\ell} |\ell\rangle$$

$$\text{select}(H) = \sum_{\ell=1}^{L} |\ell\rangle\langle\ell| \otimes H_\ell$$

and construct V_ϕ as



- Qubitization [Low, Chuang 16]: if $H/\alpha = \sum_\lambda \lambda |\lambda\rangle\langle\lambda|$, then

$$V_\phi = \sum_{\lambda_\pm} e^{\frac{i}{2}\theta_{\lambda_\pm}} R_\phi(\theta_{\lambda_\pm}) \otimes |\lambda_\pm\rangle\langle\lambda_\pm|$$

with rotation angles

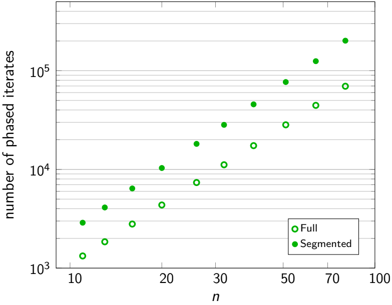
$$\theta_{\lambda_+} = \arcsin(\lambda) + \pi$$
$$\theta_{\lambda_-} = -\arcsin(\lambda)$$

- Signal processing: implement sin function via

$$R_{\phi_M}(\theta) \cdots R_{\phi_1}(\theta)$$
$$= A(\cos\tfrac{\theta}{2})\, I + iB(\cos\tfrac{\theta}{2})\, \sigma_z$$
$$+ i \cos\tfrac{\theta}{2} C(\sin\tfrac{\theta}{2})\, \sigma_x + i \cos\tfrac{\theta}{2} D(\sin\tfrac{\theta}{2})\, \sigma_y$$

Segmented QSP

- The computation of phases $\phi_1, \dots, \phi_M$ is difficult in practice.
- Example: the computation becomes costly when $M \geq 32$, but error analysis suggests taking $M = 1100$ to simulate 10 qubits.
- Workarounds:
  - use placeholder values
  - divide the evolution time into segments; each has length M sufficiently small that phase angles can readily be computed
- Overhead is not too large: the segmented QSP has complexity $O(n^{3+4/M})$, compared to $O(n^3)$ for the full QSP.

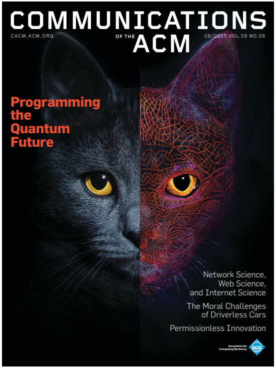


Empirical bounds

- **Rigorous bound can be loose.**
- For PF, we extrapolate from numerical simulations of systems of size 5 to 12.
- For TS, empirical bound is infeasible but probably not helpful.
- For QSP, we find an improved empirical estimate of the truncation error of the Jacobi-Anger expansion, leading to a small reduction in the gate count.
- Preliminary evidence suggests full empirical bound for QSP will probably not be helpful.

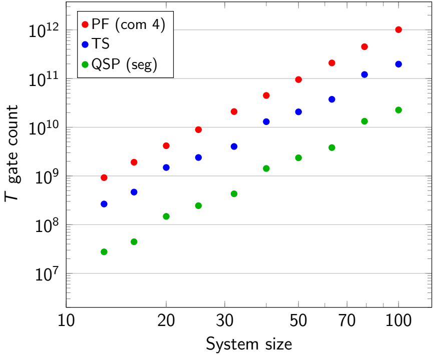
Circuit synthesis and optimization

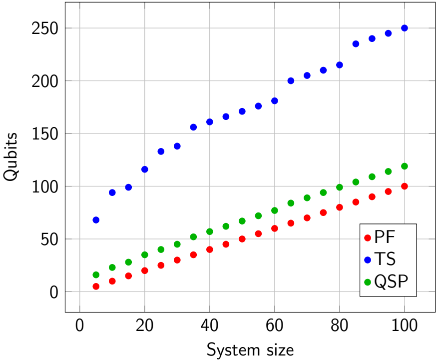
- **We implement all algorithms using Quipper,** a circuit description language facilitating concrete resource counts.
- Circuits are expressed over Clifford+$R_z(\theta)$ and Clifford+T.
- We verified correctness by simulating subroutines and small instances.
- Implementation available at `github.com/njross/simcount`



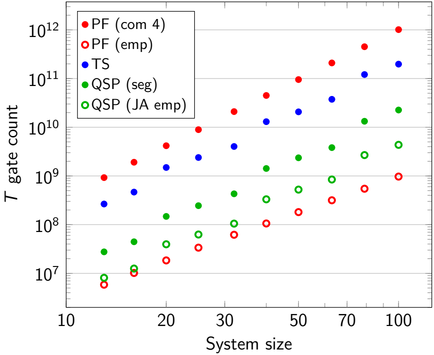
- We also applied **an automated quantum circuit optimizer [arXiv:1710.07345]** that we developed. $CNOT/T$ counts improve by about 30% for PF, less significantly for TS/QSP.

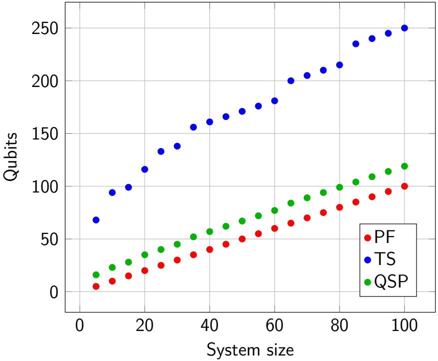
Results


PF (com 4)
TS
QSP (seg)
10^{12}
10^{11}
10^{10}
10^{9}
10^{8}
10^{7}
T gate count
10
20
30
50
70
100
System size

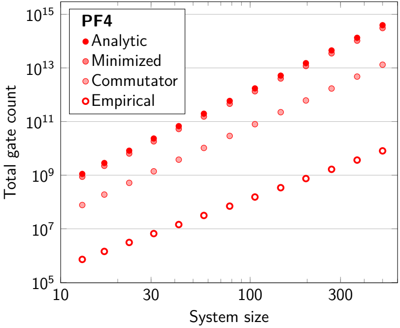

250
200
150
100
50
0
Qubits
PF
TS
QSP
0
20
40
60
80
100
System size

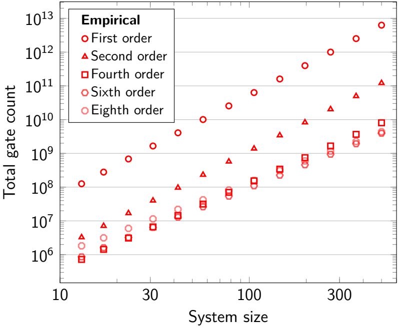
Results (the full story)





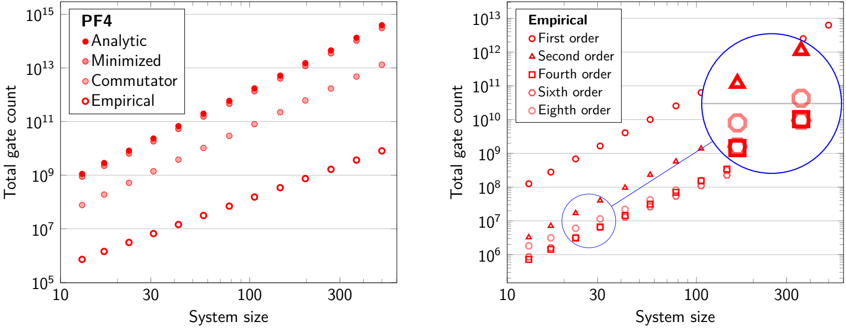
PF algorithm: orders and bounds





	Order				
Bound	1	2	4	6	8
Analytic/Minimized	5	4	3.5	3.333	3.25
Commutator	4	3.667	3.4	3.286	3.222
Empirical	2.964	2.883	2.555	2.311	2.141

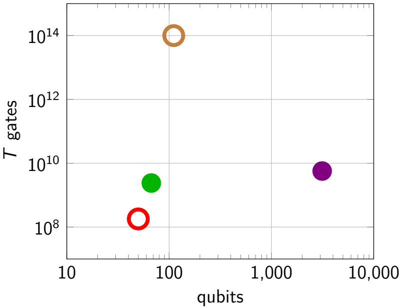
PF algorithm: orders and bounds



Bound	Order 1	2	4	6	8
Analytic/Minimized	5	4	3.5	3.333	3.25
Commutator	4	3.667	3.4	3.286	3.222
Empirical	2.964	2.883	2.555	2.311	2.141

Comparison with related work

- **Factoring a 1024-bit number [Kutin 06]**
  - 3132 qubits
  - 5.7×10^9 T gates



- **Simulating FeMoco [Reiher et al. 16]**
  - 111 qubits
  - 1.0×10^{14} T gates

- **Simulating 50 spins (segmented QSP)**
  - 67 qubits
  - 2.4×10^9 T gates

- **Simulating 50 spins (empirical PF)**
  - 50 qubits
  - 1.8×10^8 T gates

Summary

- This work represents progress toward the first genuine application of quantum computers, solving a practical problem that is beyond the reach of classical computation.

- Spin models are much easier than factoring or quantum chemistry, but may still be out of reach of pre-fault tolerant devices.

- Useful takeaways:
  - Higher-order PFs are useful even for very small systems.
  - More sophisticated algorithms (especially QSP) give the best performance with rigorous guarantees at surprisingly small sizes.
  - Existing PF error bounds are very loose.

Future studies

- Better provable performance for simulation algorithms
  - Closing the gap between rigorous and empirical PF
  - Efficient synthesis of full QSP circuit
- Resource estimates for more practical models
  - Architectural constraints, parallelism
  - Different gate set
  - Fault-tolerant implementations
- Useful super-classical quantum simulation without fault tolerance?
  - Alternative target systems
  - New simulation algorithms
  - Experiments!

"Theory is the first term in the Taylor series of practice."

— Thomas M. Cover

Second-order commutator bound

Second-order commutator bound, succinct form

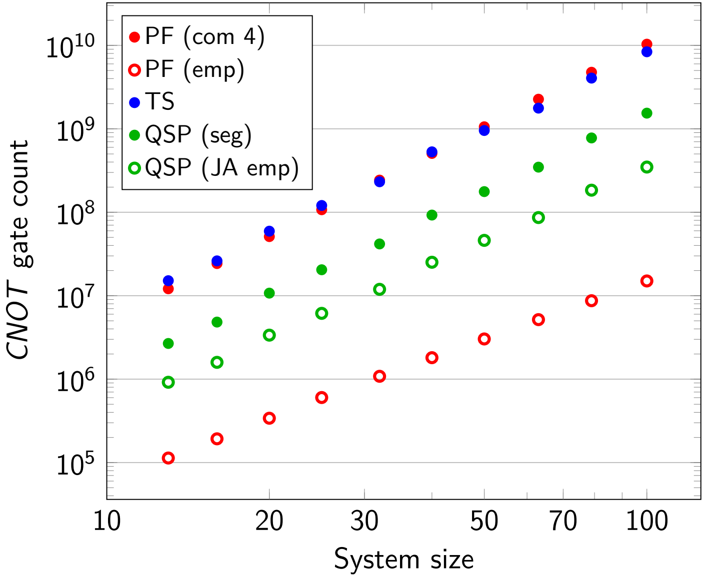
Let H be the one-dimensional nearest-neighbor Heisenberg model with a random magnetic field in the z direction. Then the error in the second-order product formula approximation satisfies

$$\begin{aligned} &\left\|\exp(-iHt) - \left[S_2\left(-it/r\right)\right]^r\right\| \\ \leq &\frac{|t|^3}{r^2} T_2(n) + \frac{4(4nt)^4}{3r^3} \exp\left(\frac{8n|t|}{r}\right), \end{aligned}$$

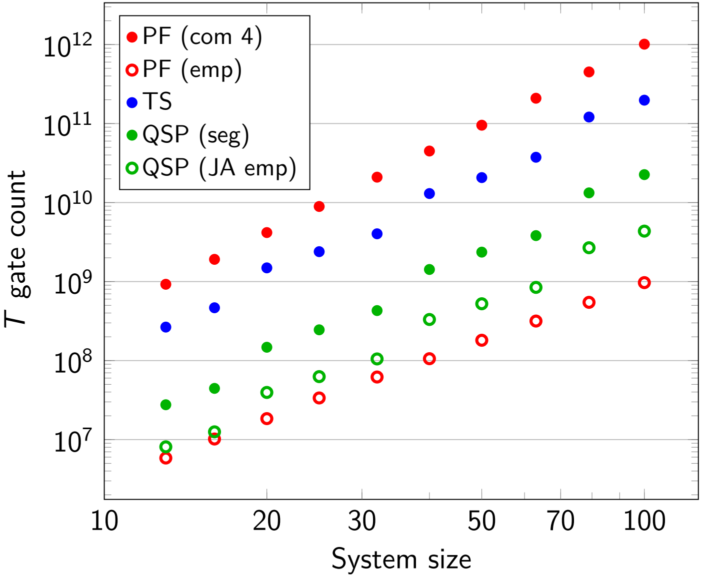
where

$$T_2(n) := \begin{cases} 194, & n = 3 \\ 40n^2 - 58n, & n \geq 4. \end{cases}$$

CNOT count

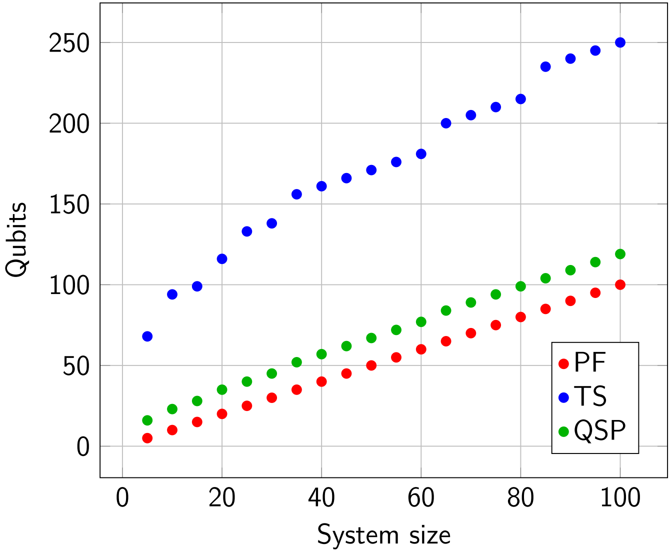

PF (com 4)
PF (emp)
TS
QSP (seg)
QSP (JA emp)
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
10^{5}
CNOT gate count
10
20
30
50
70
100
System size

T count



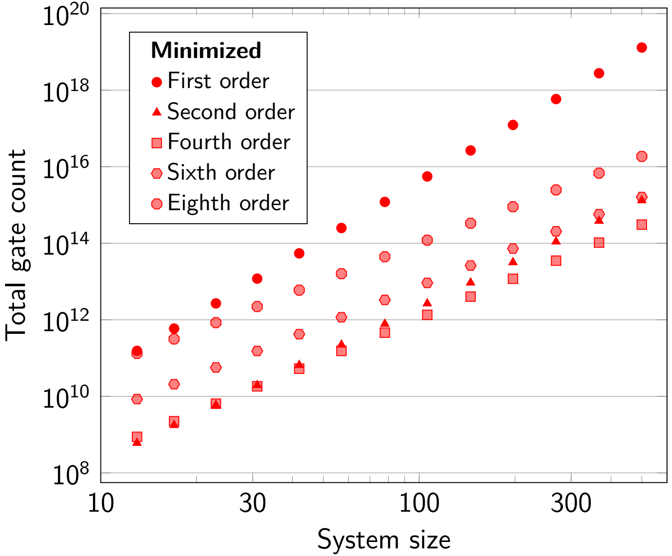
PF (com 4)
PF (emp)
TS
QSP (seg)
QSP (JA emp)
T gate count
10^{12}
10^{11}
10^{10}
10^{9}
10^{8}
10^{7}
10
20
30
50
70
100
System size

Qubit count

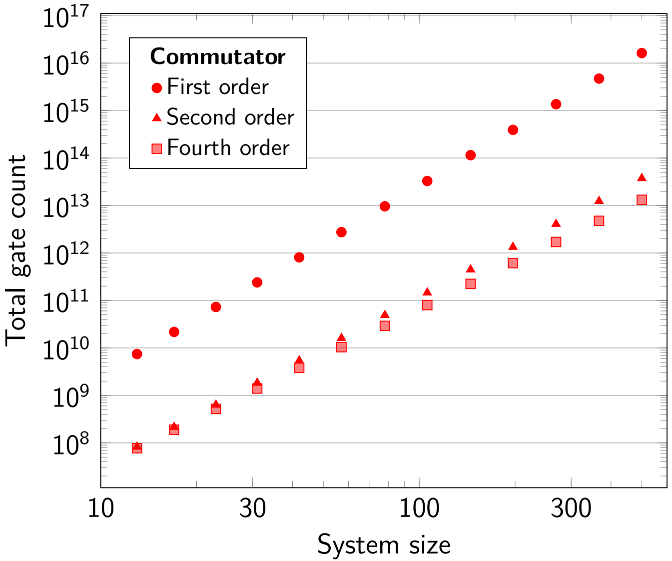


Qubits
System size
0
50
100
150
200
250
0
20
40
60
80
100
PF
TS
QSP

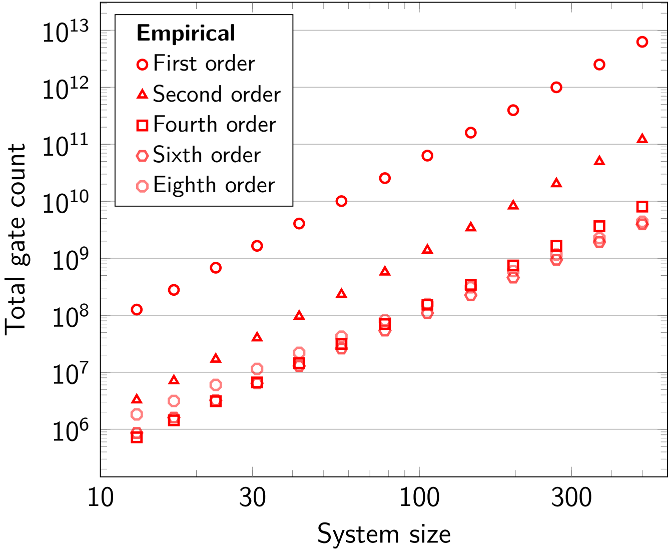
Total gate count for PF



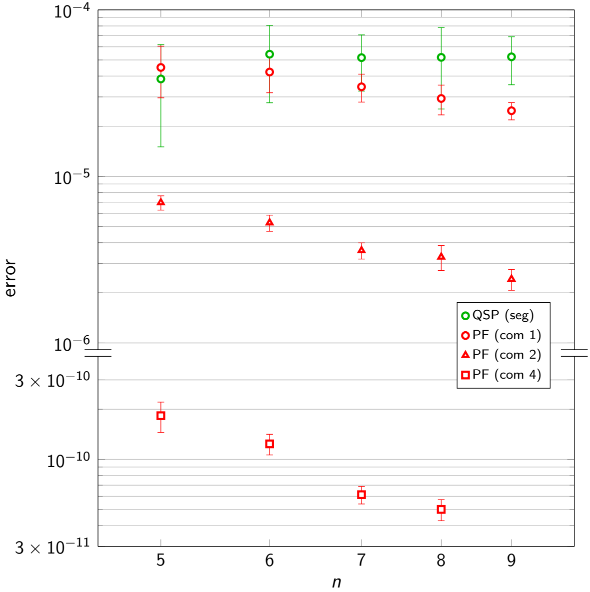
Total gate count for PF



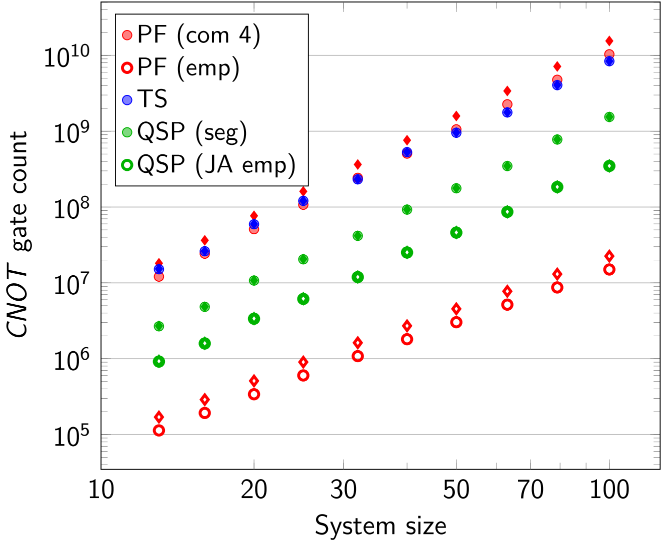
Total gate count for PF



Empirical data for QSP



CNOT optimization



T optimization