Title: Asymptotically Optimal Topological Quantum Compiling

Date: Oct 30, 2013 04:00 PM

URL: http://pirsa.org/13100129

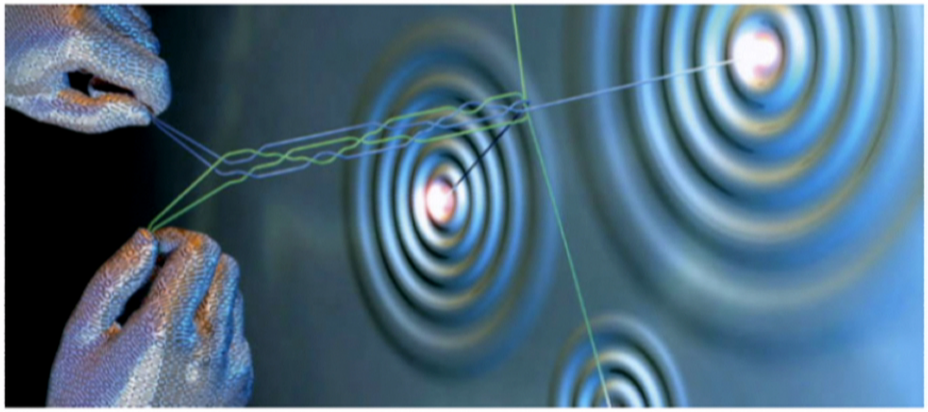
Abstract: In a topological quantum computer, universality is achieved by braiding and quantum information is natively protected from small local errors. We address the problem of compiling single-qubit quantum operations into braid representations for non-abelian quasiparticles described by the Fibonacci anyon model. We develop a probabilistically polynomial algorithm that outputs a braid pattern to approximate a given single-qubit unitary to a desired precision. We also classify the single-qubit unitaries that can be implemented exactly by a Fibonacci anyon braid pattern and present an efficient algorithm to produce their braid patterns. Our techniques produce braid patterns that meet the uniform asymptotic lower bound on the compiled circuit depth and thus are depth-optimal asymptotically. Our compiled circuits are significantly shorter than those output by prior state-of-the-art methods, resulting in improvements in depth by factors ranging from 20 to 1000 for precisions ranging between 10^{âˆ’10} and 10^{âˆ’30}.

Outline

- Motivation
- Model
  - Requirements
  - Hilbert space
- Single qubit operations
- Number theoretic side
  - Exact synthesis
  - Approximation
- Experiments

Why topological quantum computers are so cool?

- Topological quantum computation:
  - *A. Yu. Kitaev* Fault-tolerant quantum computation by anyons
  - *M. H. Freedman, M. Larsen, and Z. Wang,* A modular functor which is universal for quantum computation
- Fibonacci anyon model is intrinsically **fault tolerant and universal**



Why compile for Fibonacci anyons?

[practical side]

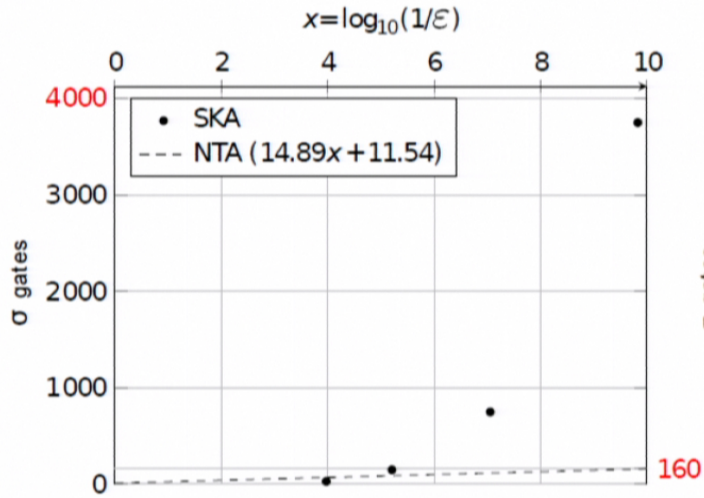
- The best known way so far: the Solovay-Kitaev algorithm
  - Gives $O\left(\log^{3+\delta}(1/\varepsilon)\right)$ instead of $O(\log(1/\varepsilon))$
  - The goal is to get an efficient asymptotically optimal algorithm – saturating the lower bound
- Implementation proposal:
  - *Roger S. K. Mong et al* Universal topological quantum computation from a superconductor/Abelian quantum Hall heterostructure

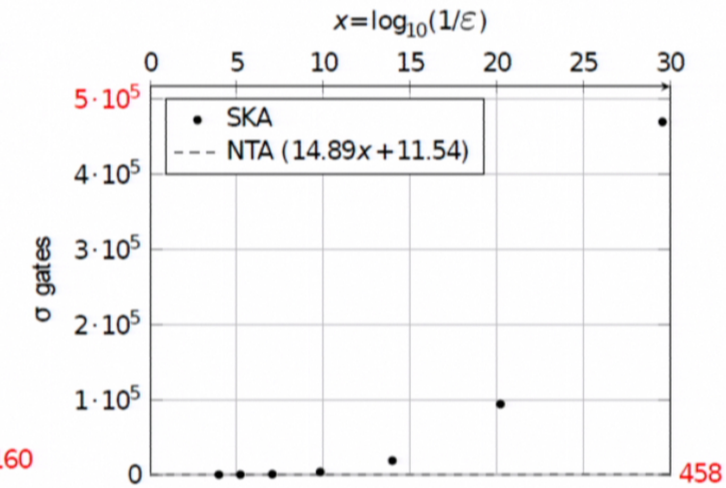
Why compile for Fibonacci anyons?

- Compiling is a much more common task for this model than for Clifford+T
  - (will see why)
- The interesting example in addition to Clifford+T [1-3] and Clifford+V [4]
  - Get better understanding what ingredients we need to solve larger classes of compilation problems

[1] *V Kliuchnikov, D Maslov, M Mosca* **Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates**
[2] *P Selinger* **Efficient Clifford+T approximation of single-qubit operators**
[3] *V Kliuchnikov, D Maslov, M Mosca* **Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits**
[4] *A Bocharov, Y Gurevich, K M. Svore* **Efficient Decomposition of Single-Qubit Gates into** V **Basis Circuits**

New method vs the Solovay-Kitaev algorithm





What we need to know?

- Given: N quasi-particles on the plane
- Two things we need to work with a system:
  - State space for the system
  - Description of the operations on the system
- Questions:
  - How to construct Hilbert space where particles live?
  - How to compute unitaries corresponding to operations on particles?

What is different ?

(from things we are used to in quantum computing)

- Hilbert space for two particles is not a tensor product of Hilbert spaces for each of them
- Dimension scales exponentially with the number of particles, but as $O(\varphi^N)$
  $\varphi = \frac{\sqrt{5}+1}{2}$

Questions and goals

Questions

- How to embed computational Hilbert space into Hilbert space describing N anyons?
- How to translate operations on anyons into operations on qubits?

Goals

- Define computational basis.
- Be able to compute unitaries acting on computational basis resulting from operations on anyons.

Types of particles and fusion rules

- Two types
  - 0 – no particle
  - 1 – the only particle in the model
- Fusion rule

$$1 \times 1 = 0 + 1$$

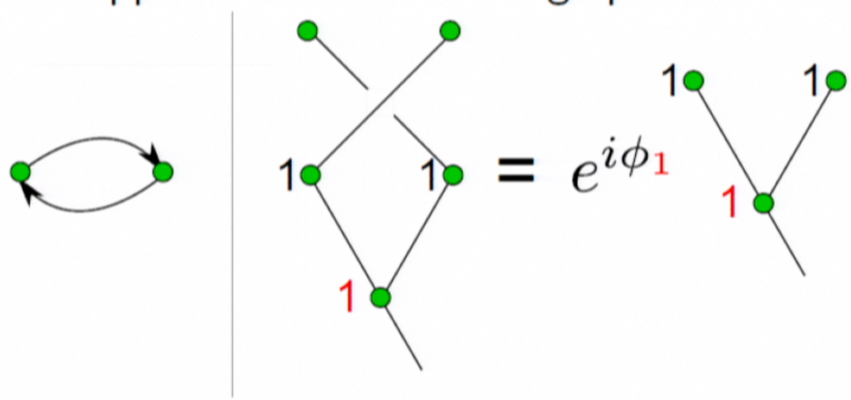
- Fusion result: type of the particle obtained by two particles combined:
  - "weird" feature: considering two anyons together you can have a charge either 0 either 1
  - To figure out what actually happened one needs to perform a measurement
- Example from usual physics :
  - fermion $\times$ fermion $=$ boson

Building Hilbert space

(describing two particles)

- Possible states

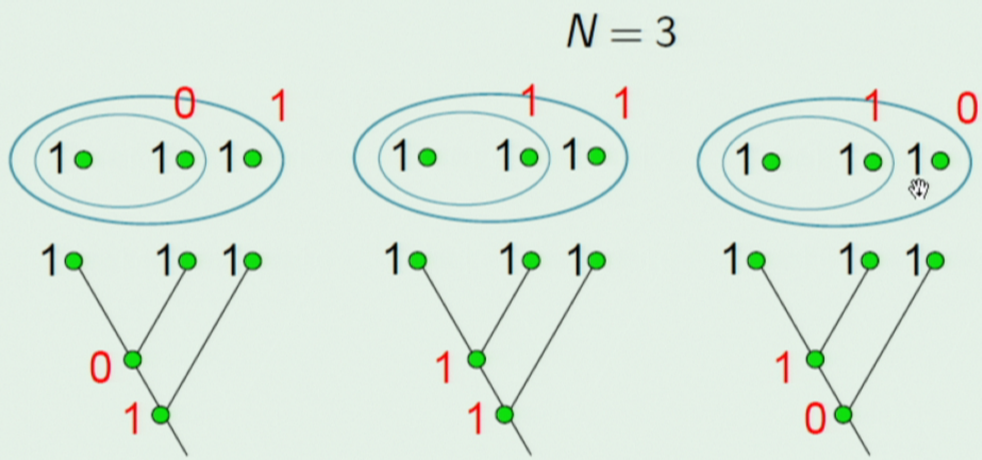
- What happens when we exchange particles?



$$\phi_0 = e^{-4\pi i/5}, \phi_1 = e^{3\pi i/5}$$

Building Hilbert space
(describing N particles)

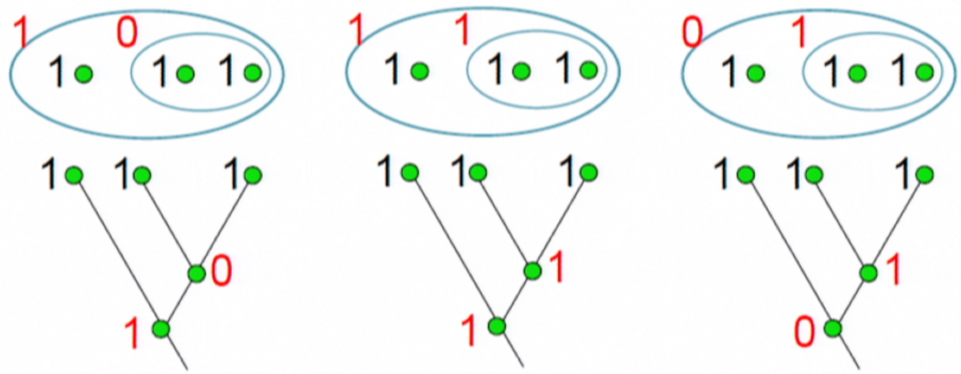
Example



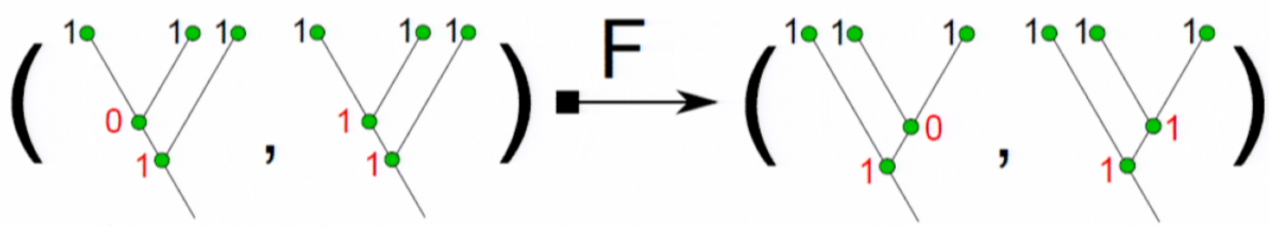
- Is this a proper basis definition?

Building Hilbert space
(describing N particles)

- What about this basis?



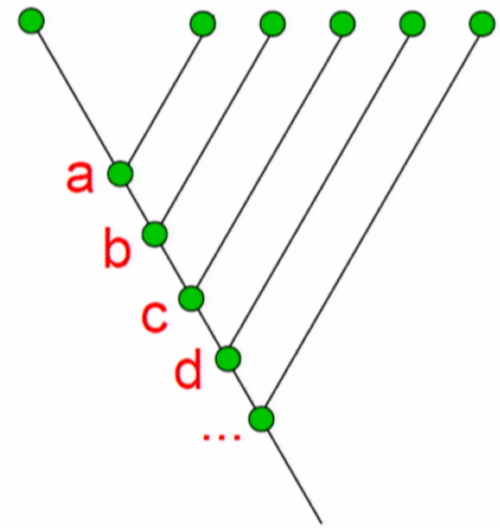
- F matrix:



$$F = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}, \tau = \frac{\sqrt{5}-1}{2}$$

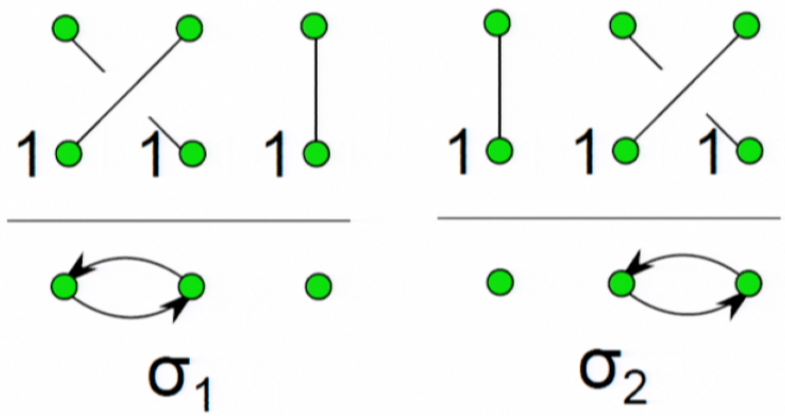
Building Hilbert space

(describing N particles)



Note: Dimension of Hilbert space scales as $O(\varphi^N)$

Available single qubit matrices



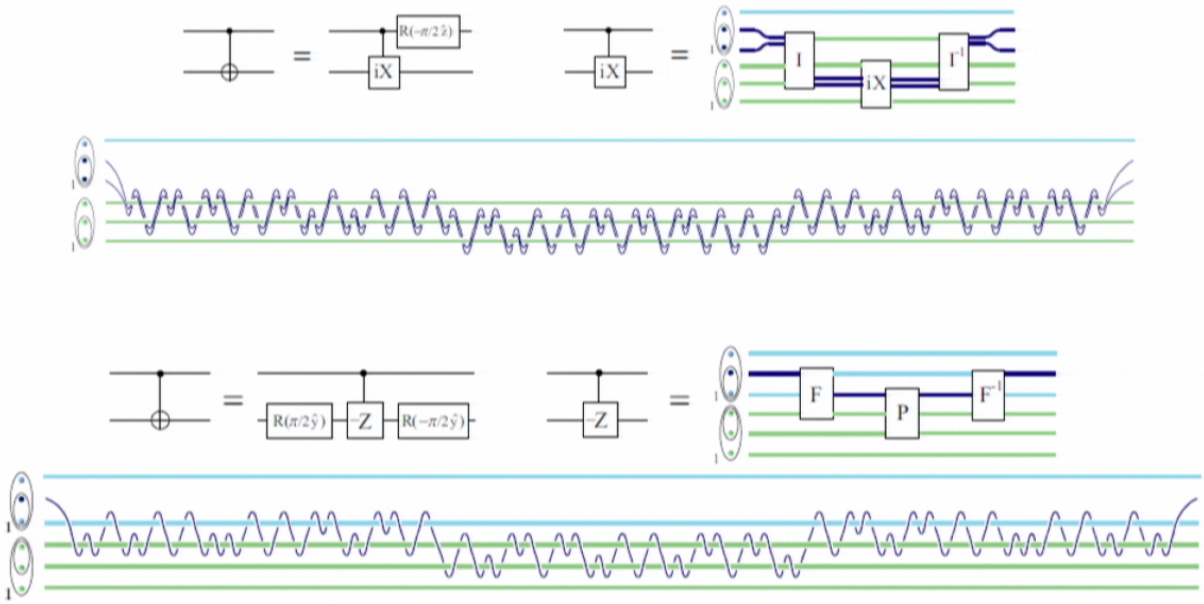
$$\sigma_1 \longleftrightarrow R = \begin{pmatrix} e^{-i\frac{4\pi}{5}} & 0 \\ 0 & e^{i\frac{3\pi}{5}} \end{pmatrix} \quad \sigma_2 \longleftrightarrow FRF,\ F = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}$$

Note

- Bit flip is not implementable exactly!

Examples of CNOT gates

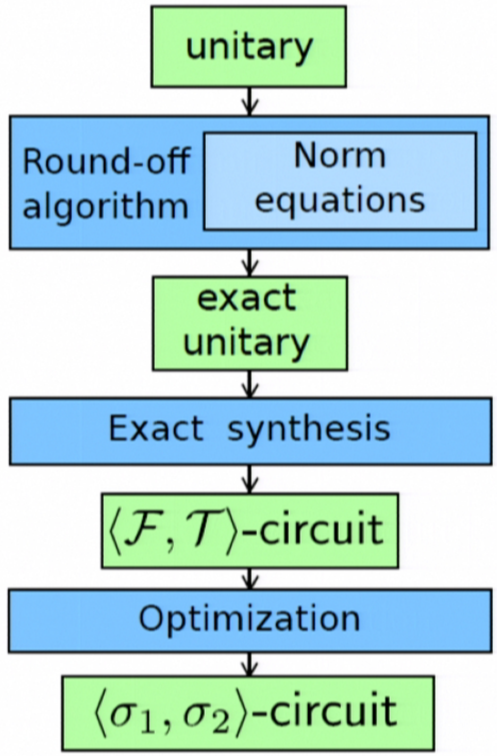
(Precision $2 \cdot 10^{-3}$ and $5 \cdot 10^{-3}$)



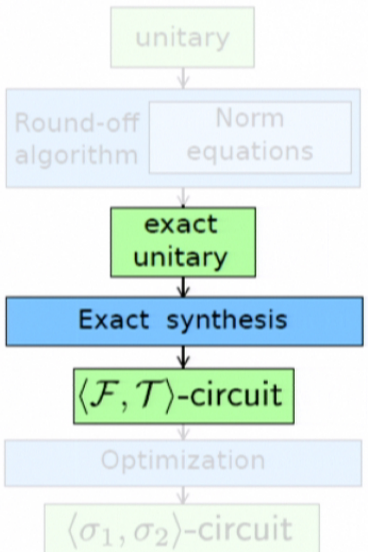
[*L.Hormozi, G.Zikos, N.E.Bonesteel, S.H.Simon*
Topological Quantum Compiling]

Compiling using number theory

[generic scheme]

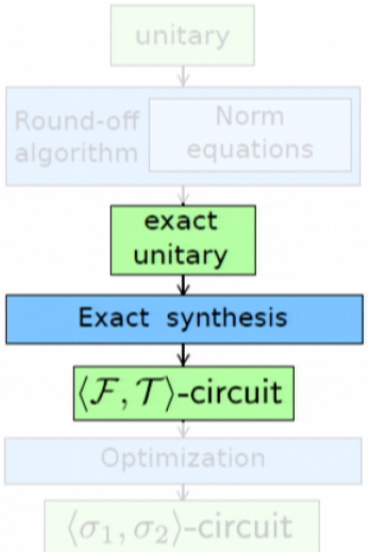


Generating matrices



- $\mathscr{T} = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}$
- $\mathscr{F} = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}$
- $\tau = \frac{\sqrt{5}-1}{2}$,
  $\omega = e^{i\pi/5}$

Exactly synthesisable unitaries



Complex ring of integers:

- $\mathbb{Z}[\omega]$:
  - $a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3$,
    a_0, a_1, a_2, a_3 – usual integers

The most general form of an exactly synthesisable unitary

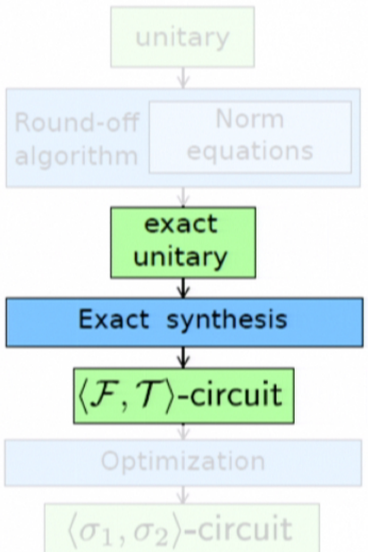
- $U[u,v,k] = \begin{pmatrix} u & -v^*\sqrt{\tau}\omega^k \\ v\sqrt{\tau} & u^*\omega^k \end{pmatrix}$
  where u, v are from $\mathbb{Z}[\omega]$
  - $|u|^2 + \tau|v|^2 = 1$

Example:

$$(\mathscr{F}\mathscr{T})^2 = \begin{pmatrix} (2-\omega+\omega^3) & -(-1+\omega-\omega^3)^*\sqrt{\tau}\omega^2 \\ (-1+\omega-\omega^3)\sqrt{\tau} & (2-\omega+\omega^3)\omega^2 \end{pmatrix}$$

Exact synthesis algorithm

[Complexity measure]



Real ring of integers:

- $\mathbb{Z}[\tau]$:
  - $a_0 + a_1\tau$, a_0, a_1 – usual integers
  - for any x from $\mathbb{Z}[\omega]$: $|x|^2$ belongs to $\mathbb{Z}[\tau]$

Galois automorphism:

- $(\tau)^\bullet = -\varphi = -(\tau + 1)$
- $(1)^\bullet = 1$

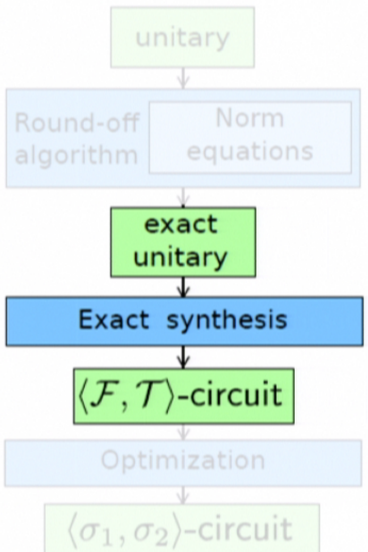
Complexity measure:

- $\mu(u) = \left(|u|^2\right)^\bullet$

Example: $\left|2 - \omega + \omega^3\right|^2 = 4 - 5\tau$,

$\mu\left(2 - \omega + \omega^3\right) = \left(\left|2 - \omega + \omega^3\right|^2\right)^\bullet = 9 + 5\tau$,

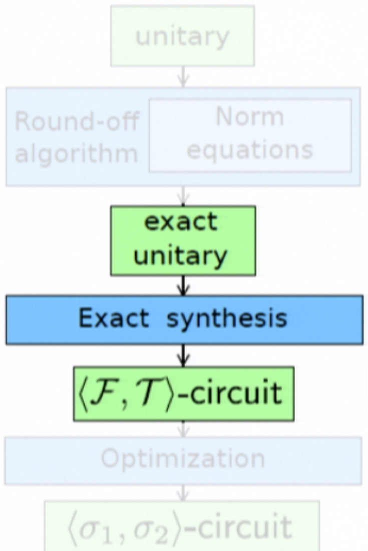
Example



$$U[u_n, v_n, k_n] = (\mathscr{F}\mathscr{T})^n$$

n	u_n	$\lvert u_n \rvert^2$	$\mu(u_n)$
0	1	1	1
1	$\omega^2 - \omega^3$	$1-\tau$	$2+\tau$
2	$2-\omega+\omega^3$	$4-5\tau$	$9+5\tau$
3	$-3+5\omega-2\omega^2-1$	$16-25\tau$	$41+25\tau$

Exact synthesis algorithm



Terminates in at most $\log_3(\mu(u))$ steps !

Input: $U[u,v,k] = \begin{pmatrix} u & -v^* \sqrt{\tau}\omega^k \\ v\sqrt{\tau} & u^*\omega^k \end{pmatrix}$

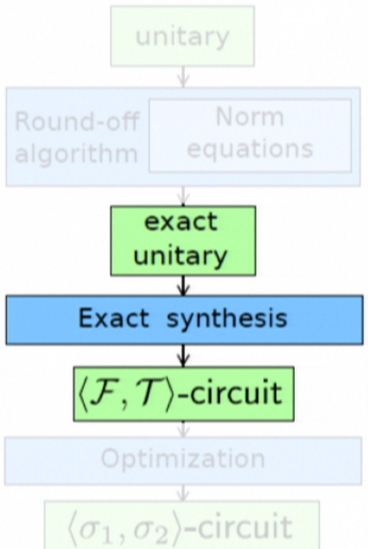
where u, v are from $\mathbb{Z}[\omega]$

- While $\mu(u) > 1$:
  - $U[u_m, v_m, k_m] = F\mathscr{T}^m U[u,v,k]$, $m = 0, \ldots, 9$
  - Pick M such that $\mu(u_M)$ is the smallest among $\mu(u_m)$
  - Add $F\mathscr{T}^{10-M}$ to the beginning of the circuit
  - $U[u,v,k] \leftarrow U[u_M, v_M, k_M]$

Note: in the end $\mu(u) = 1$ which corresponds to $u = \omega^K$, $v = 0$

Exact synthesis algorithm

[Proof outline]



Goal: Show that $\mu(u)$ can be decreased at least 3 times on each step

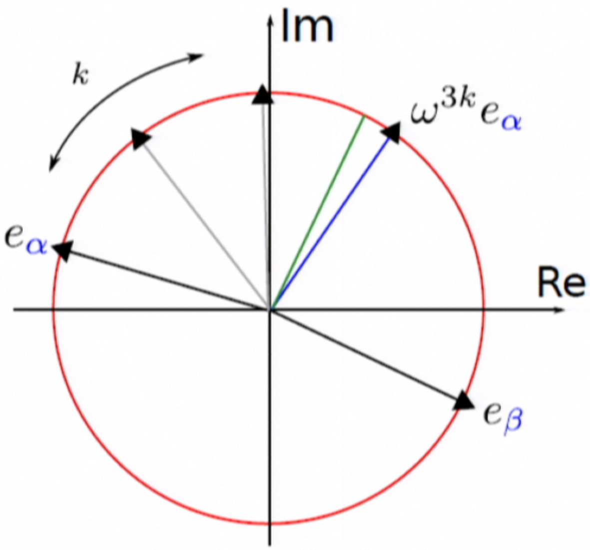
$U[u,v,k] = \begin{pmatrix} u & -v^*\sqrt{\tau}\omega^k \\ v\sqrt{\tau} & u^*\omega^k \end{pmatrix}$ where u, v are from $\mathbb{Z}[\omega]$

- $\mu(x)/\mu(v) \sim \varphi$
- For $U[u_m, y_m, k_m] = F\mathscr{T}^m U[u,v,k]$, $m = 0, \ldots, 9$
  - $\mu(u_m) = \mu(\tau(u + \omega^m v))$

Exact synthesis algorithm

[Proof outline]

Result: $\mu\left(\tau\left(u+\omega^k v\right)\right)/\mu\left(u\right)$ can be made smaller than 0.321194 by choice of k. We need at most $\log_3\left(\mu\left(u\right)\right)$ steps !!!

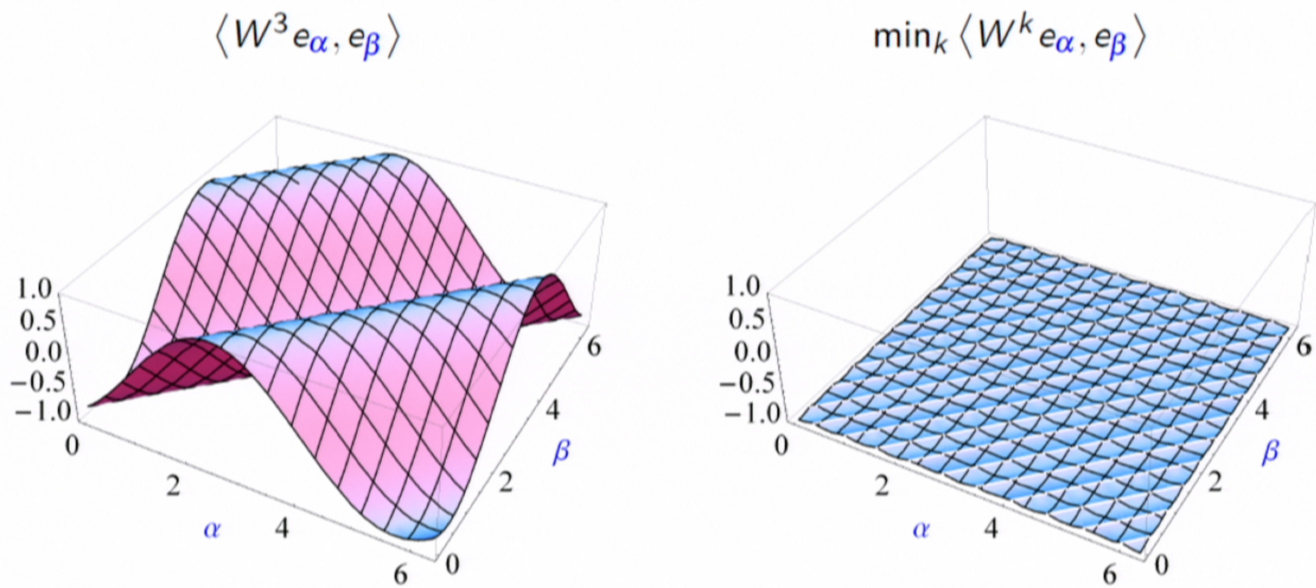


$u^\bullet = \left|u^\bullet\right| e_\alpha$ and $v^\bullet = \left|v^\bullet\right| e_\beta$

Exact synthesis algorithm

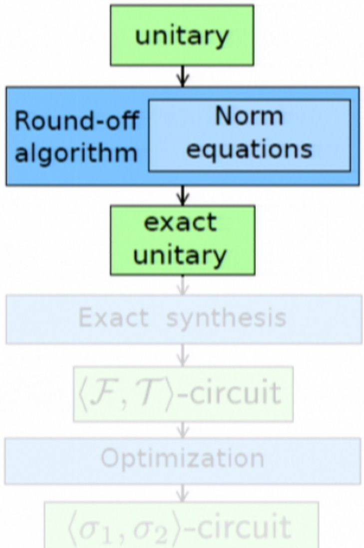
[Proof outline]

Result: $\mu\left(\tau\left(u+\omega^k v\right)\right)/\mu(u)$ can be made smaller than 0.321194 by choice of k. We need at most $\log_3(\mu(u))$ steps !!!



$\mu\left(\tau\left(u+\omega^k v\right)\right)/\mu(x) \sim \varphi^2\left(\varphi + 2\sqrt{\tau}\langle W^k e_\alpha, e_\beta\rangle\right)$
where $u^\bullet = |u^\bullet|\, e_\alpha$ and $v^\bullet = |v^\bullet|\, e_\beta$

Approximating unitaries with exactly synthesisable ones



Important special cases:

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

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

General case:

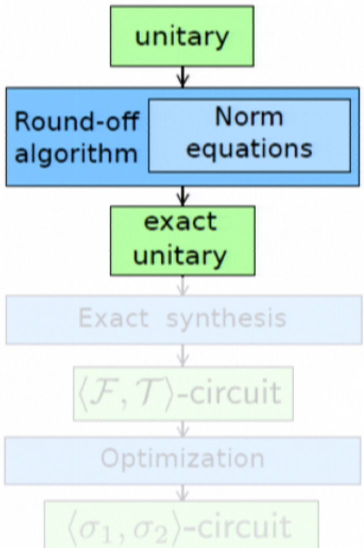
any unitary can be represented as either

$R_z(\alpha) F R_z(\beta) F R_z(\gamma)$ or $R_z(\phi)X$

(up to a global phase)

Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Target function]



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

Distance:

$$d(U, R_z(\phi)) = \sqrt{1 - \left| tr\left(U R_z^\dagger(\phi) \right) \right| /2}$$

For $U = U[u, v, 0]$:

$$d(U, R_z(\phi)) = \sqrt{1 - \left| Re\left(x e^{i\phi/2} \right) \right|}$$

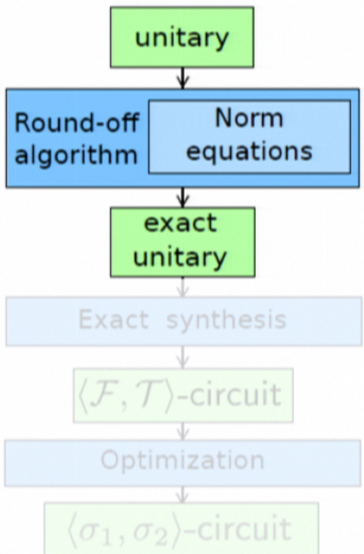
Constraints:

u, v from $\mathbb{Z}[\omega]$,

$|u|^2 + |v|^2 \tau = 1$

Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Constraints]



Relative norm equation:

$$|v|^2 = A + B\tau = \varphi\left(1 - |u|^2\right)$$

between $\mathbb{Z}[\omega]$ and $\mathbb{Z}[\tau]$

Key observation:

$$|v_1|^2 \cdot |v_2|^2 = |v_1 \cdot v_2|^2$$

Questions:

How to factor $A + B\tau$?

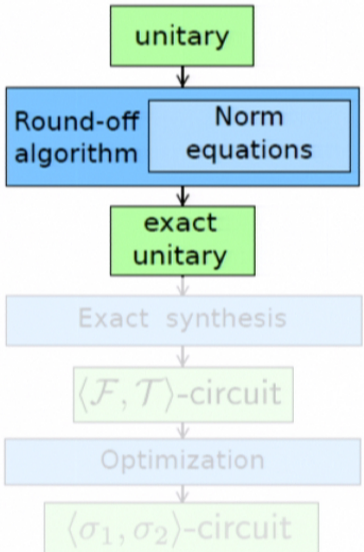
What is a "prime" number?

Tool:

$$N_\tau(A + B\tau) = A^2 - AB - B^2$$

Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Constraints: easy to find and easy to solve case]



Relative norm equation:

$$|v|^2 = A + B\tau = \varphi\left(1 - |u|^2\right)$$

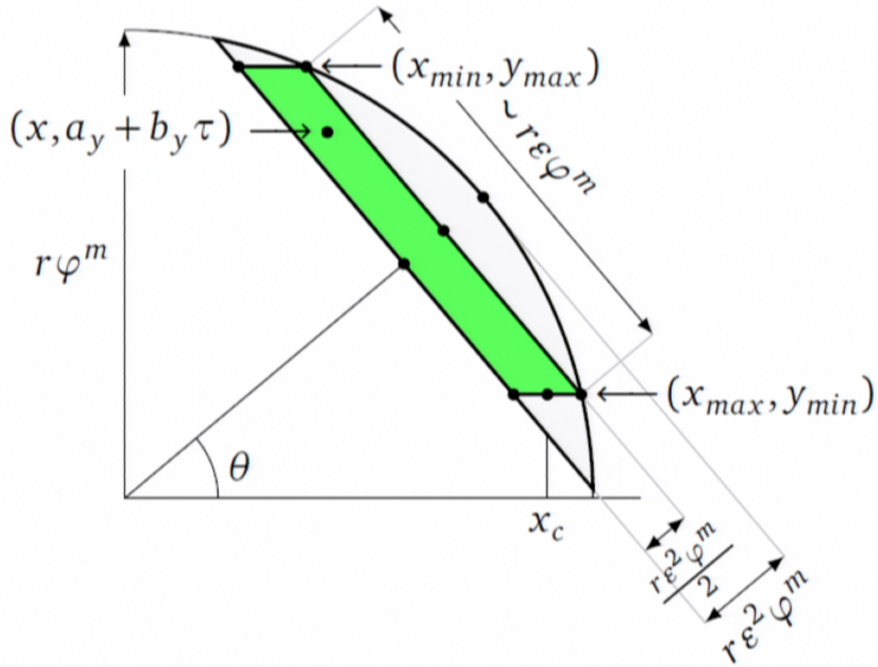
between $\mathbb{Z}[\omega]$ and $\mathbb{Z}[\tau]$

- $N_\tau(A + B\tau) = A^2 - AB - B^2$ – multiplicative function
- $a + \tau b$ – prime if $N_\tau(a + b\tau)$ is a prime number
- Easy case: $N_\tau(A + B\tau) = 5^R p$, p – prime of the from $5n + 1$
- Require: $A + B\tau > 0, (A + B\tau)^\bullet > 0$

Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Sampling stage]

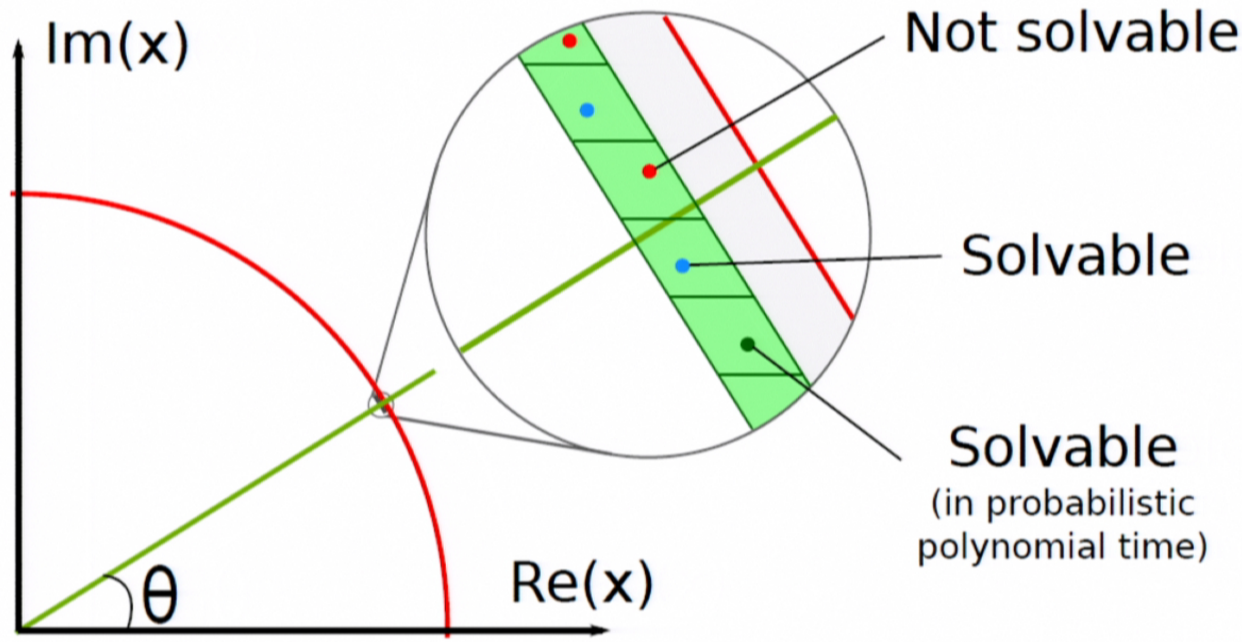
Assume $d(U[u,v,0], R_z(\phi)) \leq \varepsilon$



a_y, b_y – integer numbers defining imaginary part of u,
$m \sim \log_\tau(1/\varepsilon) + C$

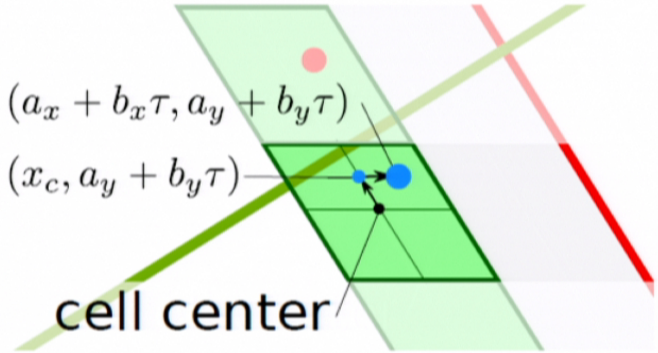
Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Sampling stage]



Approximating $R_z(\phi)$ with exactly synthesisable unitaries

[Sampling stage: choosing points]



$$\tau = \cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\dots}}}}$$

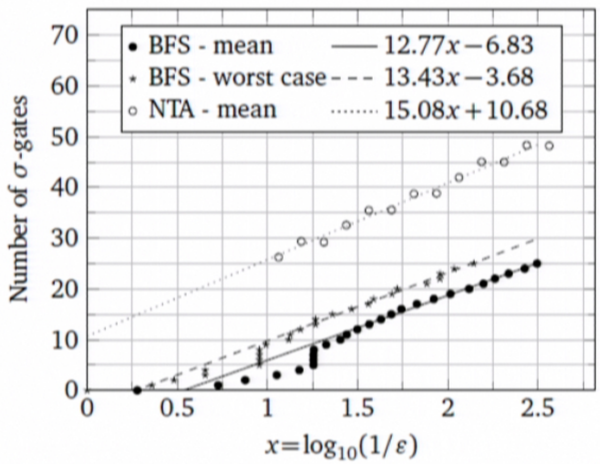
Fact

For any real x there exist integers a, b such that $|x-(a+b\tau)| \leq \tau^{n-1}(1-\tau^n)$ and $|b| \leq \varphi^n$.

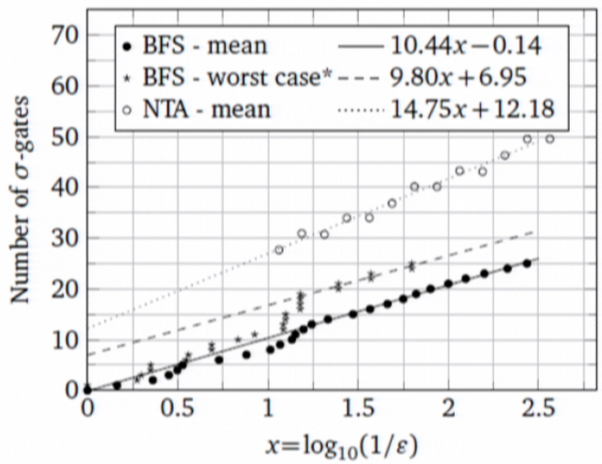
Experimental results

[Comparison to brute force search]

- C++ application
  - uses boost::multiprecision
  - uses PARI/GP to solve norm equations



$R_z(\phi)$

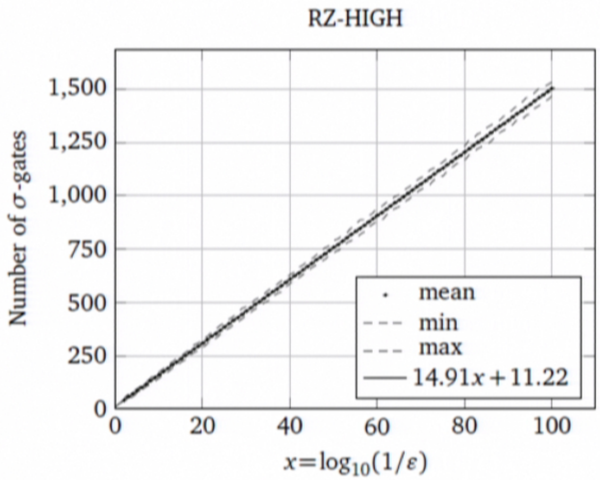


$R_z(\phi)X$

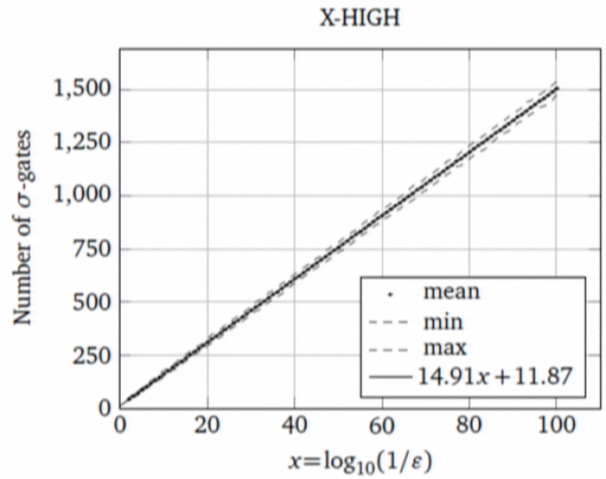
Experimental results

[High precision]

- C++ application
  - uses boost::multiprecision
  - uses PARI/GP to solve norm equations



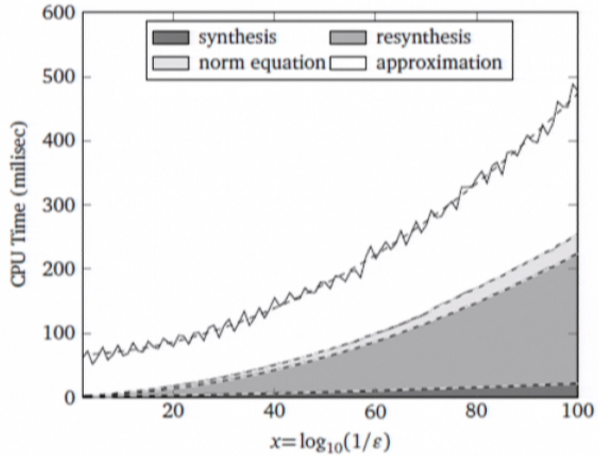
$R_z\left(\frac{\pi}{2^k}\right)$

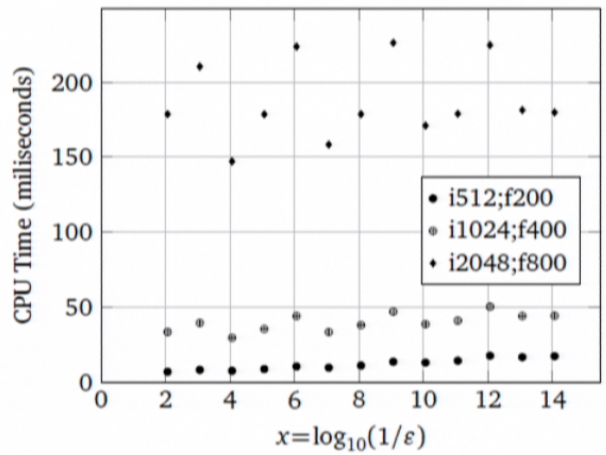


X

Experimental results

[Runtime]

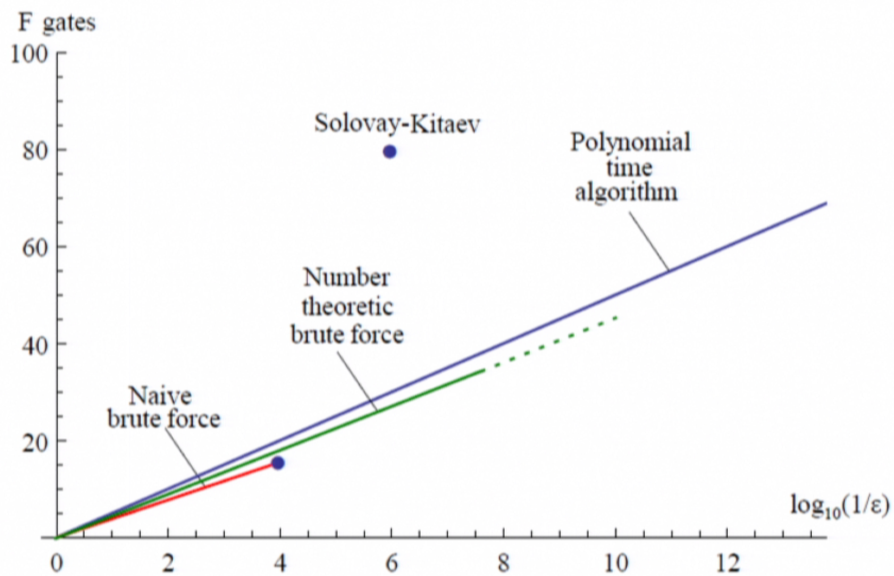




- Synthesis : $0.7539 + 0.0205x^{1.49865}$
- Resynthesis : $1.3384 + 0.0289x^{1.92089}$
- Norm equation : $1.7948 + 0.0071x^{1.80538}$
- Approximation : $62.2815 + 0.0237x^{1.90983}$

Conclusions

- Main results:
  - Efficient exact synthesis algorithm 
  - Approximation algorithm saturating the lower bound
  - φCT – compiler of single qubit operations into native instructions



Thank you

(τ from Seattle Art Museum)