Simulating quantum computers with probabilistic methods

Maarten Van den Nest
Max Planck institute for quantum optics
Garching, Germany

Vancouver, July 2010

Motivation

Why study classical simulations?

- There is a lot we don't know about the following problems:

  What are the essential ingredients responsible for quantum computational power?
  Are quantum computers truly exponentially more powerful than classical ones?
  Except quantum simulation, what would we actually do with a quantum computer if it were built?

- Two complementary routes towards understanding such questions:

  Quantum algorithms ⟺ Classical simulations

Why study classical simulations?

- There is a lot we don't know about the following problems:

What are the essential ingredients responsible for quantum computational power?
Are quantum computers truly exponentially more powerful than classical ones?
Except quantum simulation, what would we actually do with a quantum computer if it were built?

- Two complementary routes towards understanding such questions:

Quantum algorithms ⟺ Classical simulations

Why probabilistic simulations?

- Quantum mechanics is probabilistic

Outline

I. Fundamental concepts: what is classical simulation of QC?

II. Main result: class of simulatable quantum computations

III. Applications & examples

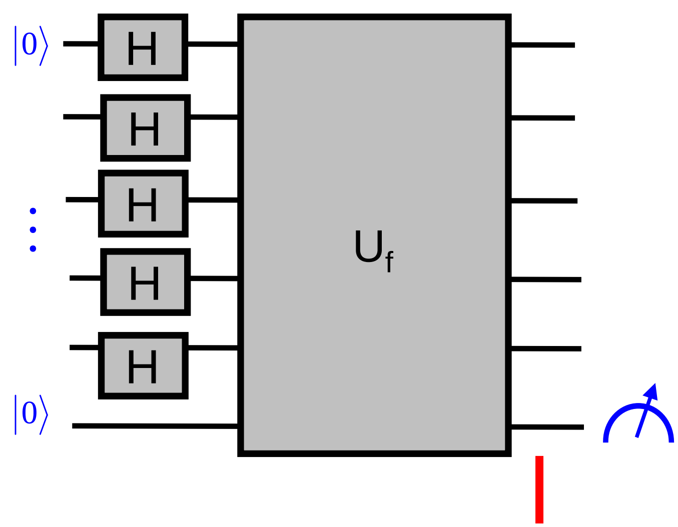
IV. Quantum algorithms

I.

Fundamental concepts

Classical simulation of QC

- Example: consider the following class of elementary quantum circuits:



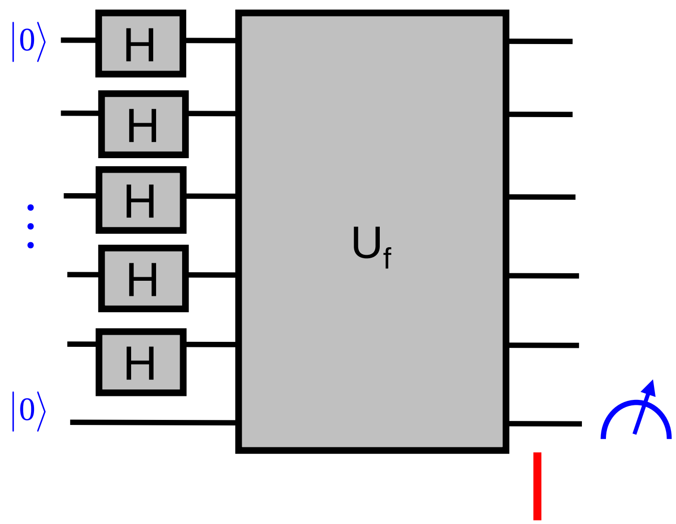
$$|\psi_{out}\rangle = \sum_x |x, f(x)\rangle$$

- Let's try to simulate this quantum circuit classically -- but what do we really mean by 'simulation'?

Strong simulation

- Classical-simulation-definition-Nr. 1: STRONG SIMULATION

> A quantum computation can be simulated classically if there exists a poly-time classical algorithm that computes $\langle \psi_{out} | Z | \psi_{out} \rangle$ with high accuracy [say, up to m bits in poly(n, m) time]



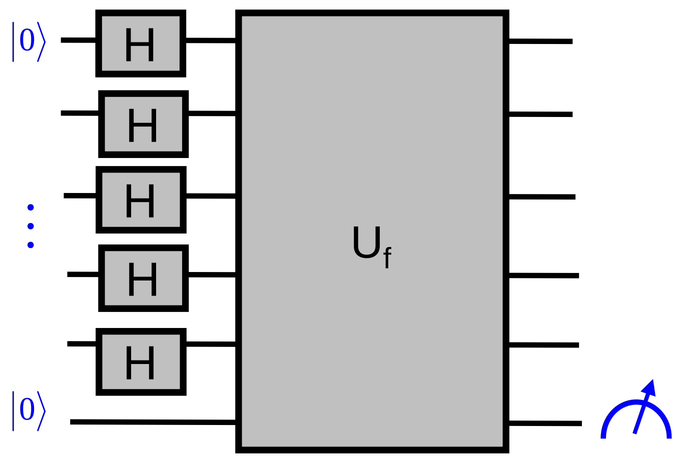
$$\langle \psi_{out} | Z | \psi_{out} \rangle = 2^{-n} \sum_x (-1)^{f(x)}$$

$$|\psi_{out}\rangle = \sum_x |x, f(x)\rangle$$

Strong simulation

- Classical-simulation-definition-Nr. 1: STRONG SIMULATION

> A quantum computation can be simulated classically if there exists a poly-time classical algorithm that computes $\langle \psi_{out} | Z | \psi_{out} \rangle$ with high accuracy [say, up to m bits in poly(n, m) time]



$$\langle \psi_{out} | Z | \psi_{out} \rangle = 2^{-n} \sum_x (-1)^{f(x)}$$

- Need to compute $|\{x : f(x) = 0\}|$ i.e. #P-complete -- so this is an un-simulatable circuit?

What's the problem?

- Consider again $|0\rangle^n \to U|0\rangle^n \equiv |\psi_{out}\rangle$ followed by Z measurement.

Q: How does this quantum computation allow to compute $\langle\psi_{out}|Z|\psi_{out}\rangle$?

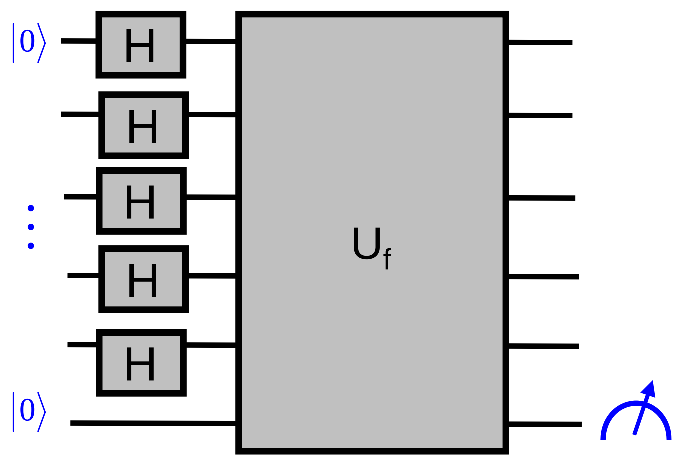
- Outcome in each run: $z_i \in \{1, -1\}$
- Repeat circuit + measurement N = poly(n) times, record outcome z_i in each run and output $c := N^{-1}\sum_i z_i$
- Then c approximates $\langle\psi_{out}|Z|\psi_{out}\rangle$ with some accuracy
- Best achievable accuracy = 1/poly(n) due to Chernoff-Hoeffding bound [with exponentially small probability of failure]

A: approximate $\langle\psi_{out}|Z|\psi_{out}\rangle$ with at most polynomial accuracy ε = 1/poly(n), i.e. up to O(log n) bits

Weak simulation

- Classical-simulation-definition-Nr. 2: WEAK SIMULATION

> The computation $|0\rangle^n \rightarrow U|0\rangle^n \equiv |\psi_{out}\rangle$ can be simulated classically if there exists a classical algorithm that approximates $\langle\psi_{out}|Z|\psi_{out}\rangle$ with 1/poly(n) accuracy in poly-time [with exponentially small failure probability]



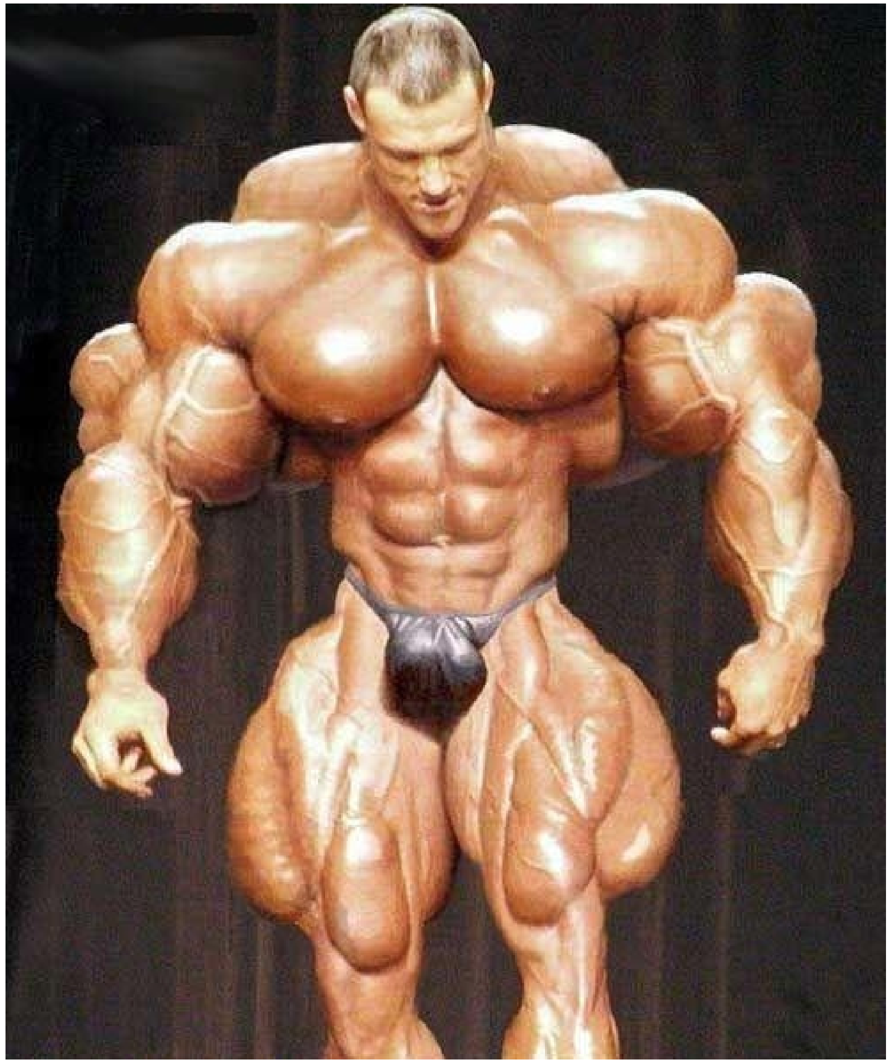
$$\langle\psi_{out}|Z|\psi_{out}\rangle = 2^{-n}\sum_x (-1)^{f(x)}$$

- Just generate N = poly(n) random bit strings x_k and output $c := \frac{1}{N}\sum_k (-1)^{f(x_k)}$

Strong versus weak simulation

The strong approach = overdoing it a bit…

Strong versus weak simulation



The strong approach = overdoing it a bit…

Our work

- Develop new weak classical simulation algorithms

- Results:
  - One main theorem
  - Several applications
  - Simulating quantum algorithms

II.

Main Theorem

CT states

- DEFINITION

An n-qubit state $|\psi\rangle$ is computationally tractable (CT) if

(a) Given x, the coefficient $\langle x|\psi\rangle$ can be computed in poly-time

(b) It is possible to sample in poly-time from $\mathrm{Prob}(x) = |\langle x|\psi\rangle|^2$

CT states

- DEFINITION

An n-qubit state $|\psi\rangle$ is computationally tractable (CT) if

(a) Given x, the coefficient $\langle x|\psi\rangle$ can be computed in poly-time

(b) It is possible to sample in poly-time from $\mathrm{Prob}(x) = |\langle x|\psi\rangle|^2$

- Examples: most of existing simulation results

  - Product states, MPS, TTN, stabilizer states, Weighted graph states
  - Standard basis inputs followed by matchgate circuits
  - Product inputs followed by Toffoli's, QFT, log-depth n.n. circuits, . . .

Overlaps of CT states

- PROPERTY: If $|\psi\rangle$ and $|\varphi\rangle$ are two CT states, then there exist an efficient classical algorithm to approximate $\langle\varphi|\psi\rangle$ with 1/poly(n) accuracy

Overlaps of CT states

- PROPERTY: If $|\psi\rangle$ and $|\varphi\rangle$ are two CT states, then there exist an efficient classical algorithm to approximate $\langle\varphi|\psi\rangle$ with 1/poly(n) accuracy

- Hint of proof:

$$\delta(x) = \begin{cases} 1 & \text{If } |\langle x|\varphi\rangle| \geq |\langle x|\psi\rangle| \\ 0 & \text{otherwise} \end{cases} \qquad \varepsilon(x) = 1 - \delta(x)$$

$$\begin{aligned} \langle\varphi|\psi\rangle &= \sum \langle\varphi|x\rangle\langle x|\psi\rangle \\ &= \sum \langle\varphi|x\rangle\langle x|\psi\rangle\delta(x) + \sum \langle\varphi|x\rangle\langle x|\psi\rangle\varepsilon(x) \\ &= \sum |\langle\varphi|x\rangle|^2 \left\{ \frac{\langle x|\psi\rangle}{\langle x|\varphi\rangle}\delta(x) \right\} \end{aligned}$$

+ [similar for $\varepsilon(x)$]

sample

Eff. Computable + bounded

CT States

- Other properties:

- PROPERTY: If $|\psi\rangle$ is a CT state and O is a d-local operator with d = O(log n), then the expectation value $\langle\psi|O|\psi\rangle$ can be estimated efficiently with 1/poly accuracy.

- PROPERTY: If $|\psi\rangle$ is a CT state and U is a poly-size circuit of Toffoli and/or diagonal gates, then $U|\psi\rangle$ is also CT.

Sparse operations

- An n-qubit operation is efficiently computable sparse (ECS) if
  - Only poly(n) nonzero entries per row/column
  - Given n-bit string x, it is possible to list all nonzero entries in row x in poly-time; similar for columns
- Examples:
  - Pauli products
  - The standard U_f operator (where f is in P)
  - A single d-qubit gate $U \otimes I$ with d = O(log n)
  - Circuits of Toffoli + diagonal (+ adding a few non-toffoli's)
  - d-local Hamiltonians

Main Theorem

- CT-THEOREM: Let $|\psi\rangle$ be an n-qubit state, let U denote a poly-size circuit and let O denote an observable. If

  (a) $|\psi\rangle$ is CT, and

  (b) $U^{\dagger}OU$ is efficiently computable sparse (ECS),

  then the circuit can be simulated efficiently [in the weak sense!]

III.

Applications

App. 1: Sparse circuits

- THEOREM: Consider an n-qubit circuit U of m gates, each of which is ECS with sparseness s, acting on a product state and followed by Z measurement. If s^m = poly(n) then this circuit can be simulated classically.

  [Proof: product state is CT + $U^{\dagger} Z_1 U$ is ECS, then use CT theorem]

- Highlights distinction between entanglement and interference in quantum computation

App. 2: "Unification"

- Consider a product input followed by poly-size circuit U that is one of the following, followed by Z measurement.
  - Clifford circuit, possibly interspersed with few non-Cliffords
  - Toffoli circuit -- i.e. "classical" computation
  - Matchgate circuit
  - Bounded-depth circuit
- All of the above cases can be simulated classically -- with very different methods!
- Product state is trivially CT + in all above cases, $U^{\dagger}ZU$ is ECS
- CT-Theorem identifies common element in these classes

App. 3: Composability

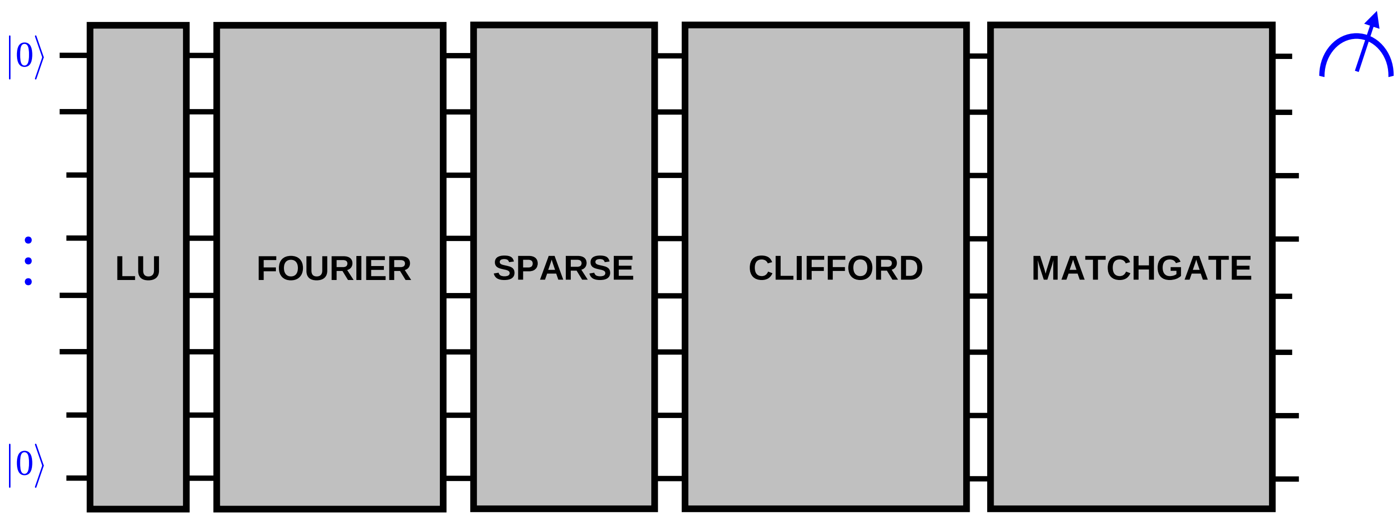
- Given two circuits U_1 and U_2 that can be simulated efficiently classically, when is the concatenated circuit U_2U_1 classically simulatable?
- Nontrivial question – cf. e.g Shor algorithm!

App. 3: Composability

- Given two circuits U_1 and U_2 that can be simulated efficiently classically, when is the concatenated circuit U_2U_1 classically simulatable?

- Nontrivial question – cf. e.g Shor algorithm!

- CT-Theorem leads to classical simulation of concatenated circuits of different types:

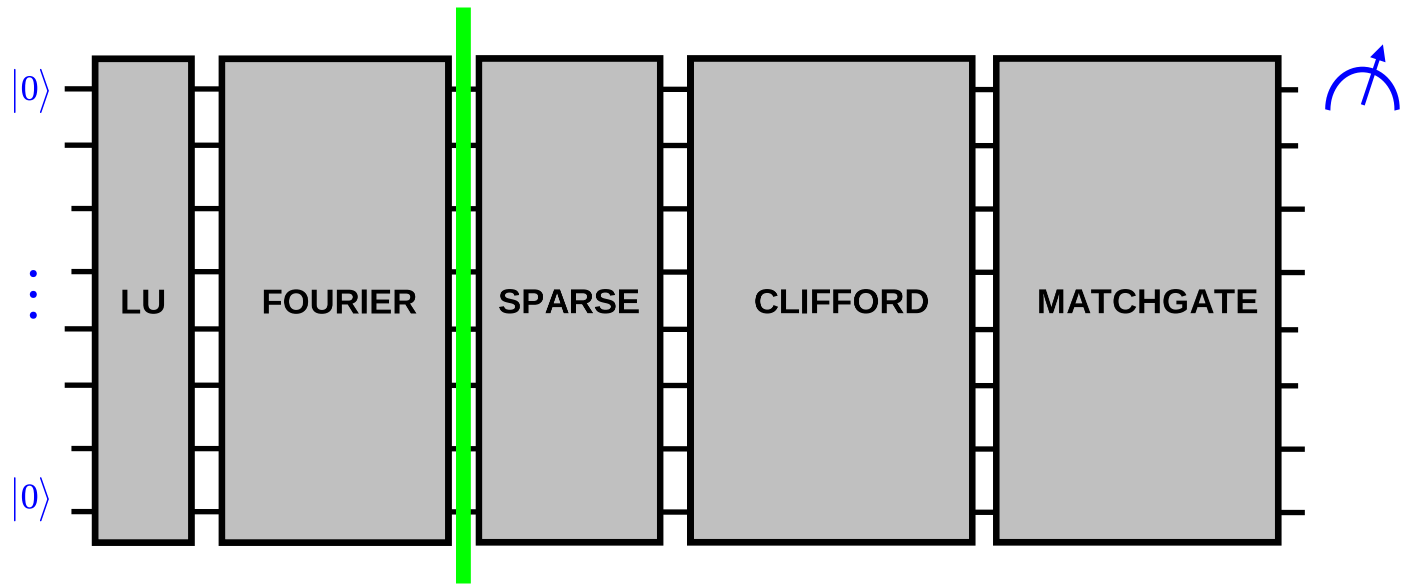


App. 3: Composability

- Given two circuits U_1 and U_2 that can be simulated efficiently classically, when is the concatenated circuit U_2U_1 classically simulatable?

- Nontrivial question – cf. e.g Shor algorithm!

- CT-Theorem leads to classical simulation of concatenated circuits of different types:



IV.

Quantum algorithms

Example #1:

Potts model q. algorithm

Potts model algorithm

- Z = partition function of classical spin system e.g. Ising, Potts, …
- I. Arad & Z. Landau '08: quantum algorithm to approximate Z/Δ with 1/poly accuracy, where Δ is easy-to-compute normalization
- VDN, Dür, Briegel '07:

$$Z/\Delta = \langle \alpha | \psi \rangle \qquad \begin{cases} |\alpha\rangle = \text{product state} \\ |\psi\rangle = \text{stabilizer state} \end{cases}$$

- Both states are CT states

  And overlaps between CT states can be estimated with 1/poly accuracy classically in poly-time

- Hence, Potts model quantum algo can be simulated classically

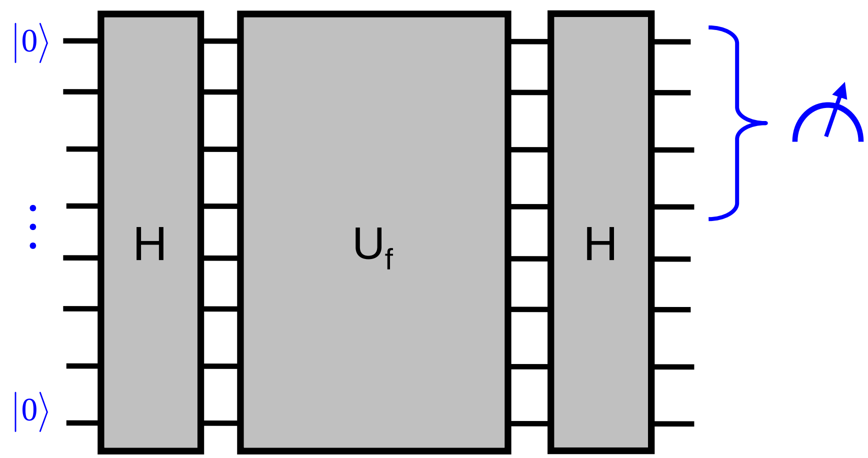
Example #2:

Simon's algorithm

Simon's algorithm

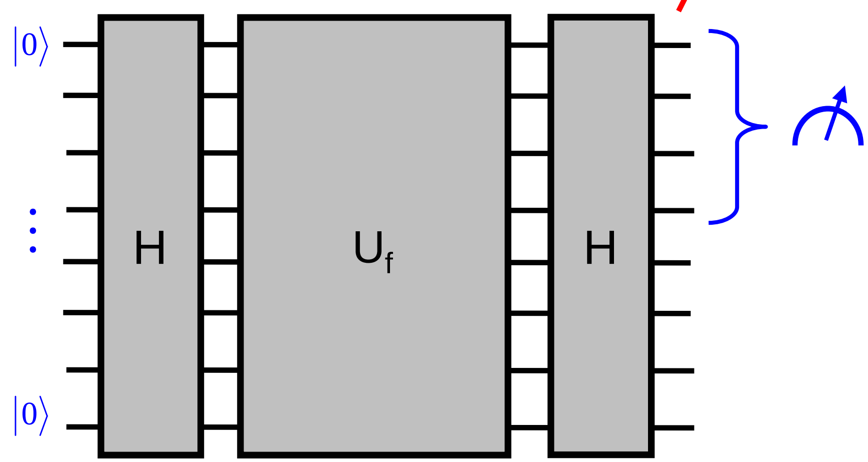
- SIMON'S PROBLEM: Consider oracle access to n-bit function f. It is promised that there exists a bit string a such that f(x) = f(y) if and only if x+y = a. Objective: find the string a.

- Classically $O(2^{n/2})$ queries are required, quantum only O(n). The quantum circuit has very simple structure:



Simon's algorithm

$$|\psi_{out}\rangle \sim \sum_{u\bullet a=0} |u\rangle|\psi_u\rangle$$

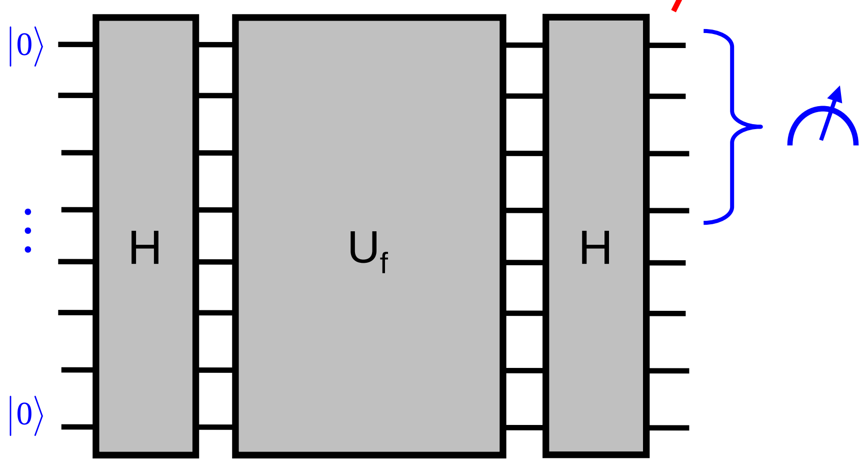


Simon's algorithm

Final step: classical postprocessing

Measure 1st register N=O(n) times, yielding N bit strings u_k that are orthogonal to a. Then solving a simple system of linear equations yields a.

$$|\psi_{out}\rangle \sim \sum_{u \bullet a = 0} |u\rangle |\psi_u\rangle$$



Simon's algorithm

- Where does the power of Simon's algorithm originate?
- Let's try to simulate such Simon-type circuits and see how far we get
- Here we focus on the -- somewhat surprising! – role of the last round i.e. classical post-processing

Simon's algorithm

- THEOREM:

  If the function computed in the classical post-processing has a sufficiently peaked Fourier spectrum, then the entire quantum computation is classically simulatable!

  i.e. nontrivial interplay between FT and classical round is required to obtain exponential speed-up!

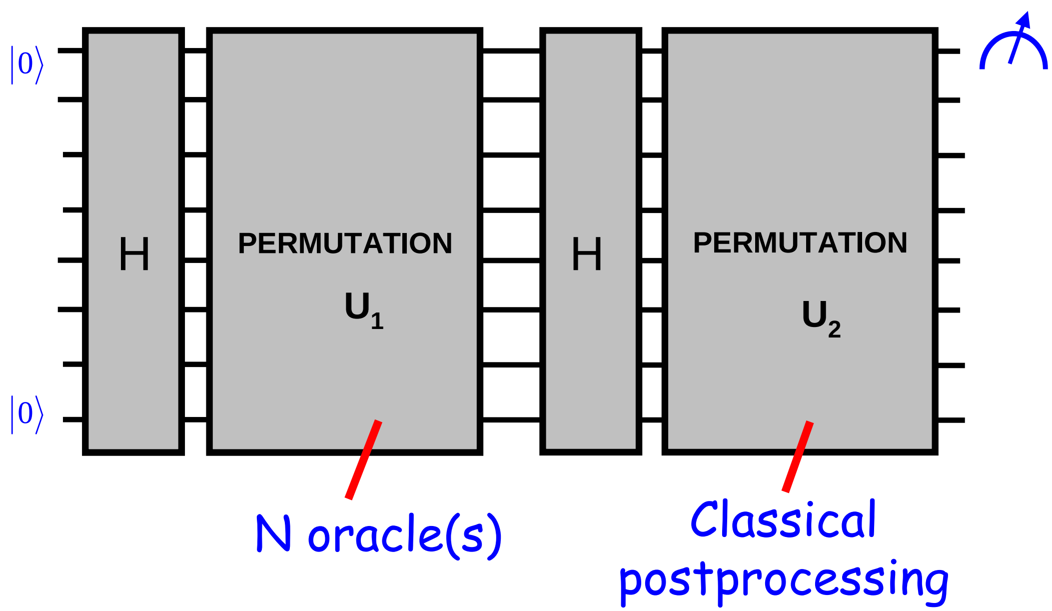
Conclusion

- Weak simulation yields new insights in simulation of QC
- We've only scratched the surface...
- See MVDN, arXiv:0911.1624
- Some advertising of new work:
  Matchgate computations and linear threshold gates
  (MVDN, arXiv:1005.1143)

Thank you very much!

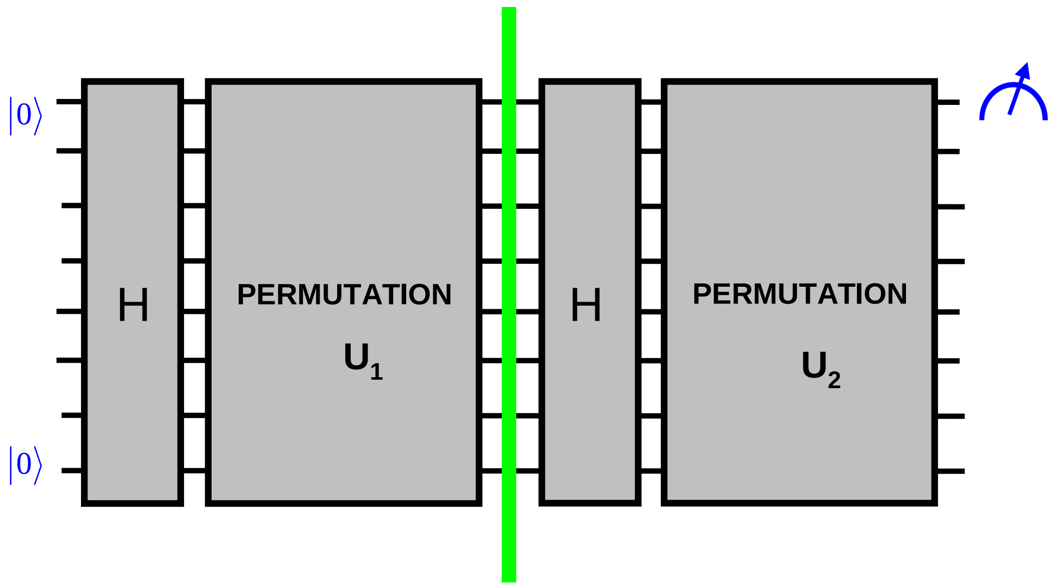
Simon's algorithm

- Doing the last round of classical computation coherently, Simon's circuit can be cast in the following form, followed by a single Z measurement



Simon's algorithm

- Doing the last round of classical computation coherently, Simon's circuit can be cast in the following form, followed by a single Z measurement



- After U_1, the system is in a CT state

- Thus, if $HU_2^{\dagger}ZU_2H$ is ECS then entire computation is simulatable – but when does this happen?

Simon's algorithm

- PROPERTY:
  Let g denote the function computed by U_2. Then the (x, y) element of $HU_2^{\dagger}ZU_2H$ equals the x+y Fourier coefficient of g, i.e.

$$\frac{1}{2^m}\sum_u (-1)^{g(u)+u^T(x+y)} = \hat{g}(x+y)$$

- If g has poly(n) non-zero Fourier coefficients then $HU_2^{\dagger}ZU_2H$ is sparse

- Nontrivial: If g has poly(n) nonzero Fourier coefficients then $HU_2^{\dagger}ZU_2H$ is well-approximated by an operator that is efficiently computable sparse

  [proof uses Kushilevitz-Mansour '93 result on learning sparse Boolean functions]

Strong versus weak simulation

- This gives an example of a class of quantum computations where
  - From the point of view of strong simulation, these quantum circuits are impossible to simulate classically (unless P = #P)
  - From the point of view of weak simulation, they are trivial to simulate classically
- Note how elementary this class of quantum circuits is (coherent version of probabilistic classical computation)