# Classical simulation of quantum computations

classically efficiently sim'ble = computationally "lame"!

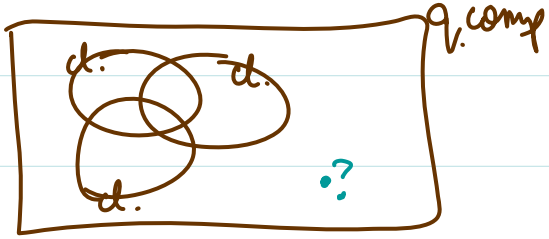
So why interesting?!

- illuminates role of quantum resources in q. computation

e.g. What added quantum-effect regains universal q. computing power?

- richer structural relationship cl → q comp.

  not just "cl ⊆ q" but many inequivalent embeddings
  - characterise them all?




- computationally easy ⇏ experimentally easy

classically sim'ble processes can involve rich quantum effects.

~ testing a q-computer? (also NP problems...)

# Notion of efficient classical simulatability

Given quantum circuit family viz. classical (eg. poly-time) computation:

$i_1 i_2 \ldots i_n \rightsquigarrow$ description of poly-sized quantum circuit on input $|0\rangle \ldots |0\rangle$ and final $Z_1$ measurement (say)

Then using only poly(n)-time classical means:

(a) (strong simulation) calculate output probabilities (to k digits of accuracy in poly(n,k) time)

(b) (weak simulation) sample output prob. distribution once.

Then can estimate output probs to $1/\text{poly}(n)$ accuracy (i.e. $O(\log n)$ digits) with prob $1-\epsilon$ for $\epsilon \sim \exp(-n)$.

(Use: Chernoff-Hoeffding — sample $k = O(n)$ times and take frequency as estimate of prob.)

This talk: we'll use strong simulation!

## Remark –

### Strong vs. weak simulation

e.g. $f$ : n-bits → 1-bit  $f(i_1 .. i_n)$ bit values: $A = \#$ 1 values.

$U_f |i_1 .. i_n\rangle |0\rangle = |i_1 .. i_n\rangle |f(i_1 .. i_n)\rangle$

$|0\rangle .. |0\rangle |0\rangle \xrightarrow{H \otimes .. \otimes H} \bullet \xrightarrow{U_f} \bullet \xrightarrow[\text{last qubit}]{\text{measure}} \bullet \quad \text{prob}(1) = \frac{A}{2^n}$

If Strong simulation – can calculate prob(1) to n digits in poly(n) time!
Hence can decide if prob(1) = 0 or not, and then get P = NP.
(Even more: can count A, and then get P = #P)

However:

Weak simulation – is easy!
Just classically choose $i_1 .. i_n$ uniformly at random
(n coin tosses) and compute $f(i_1 .. i_n)$.

# Direct simulation

Circuit = just simple linear algebra!

So calculate components of evolving state? ← by a sequence of tensor contractions.

Problem: each extra qubit ⇒ doubles dim & #components

⇒ typically exponential calc. effort with # steps!

e.g. n qubit $|\psi\rangle = \sum c_{i_1 \ldots i_n} |i_1 \ldots i_n\rangle$

Apply $V$ on qubits 1 & 2 : $V|\psi\rangle = \sum c'_{i_1 \ldots i_n} |i_1 \ldots i_n\rangle$

$$c'_{i_1 \ldots i_n} = V^{j_1 j_2}_{i_1 i_2} \, c_{j_1 j_2 i_3 \ldots i_n}$$

$i_3 \ldots i_n$ ⟷ $2^{n-2}$ strings!

But if all states are product states

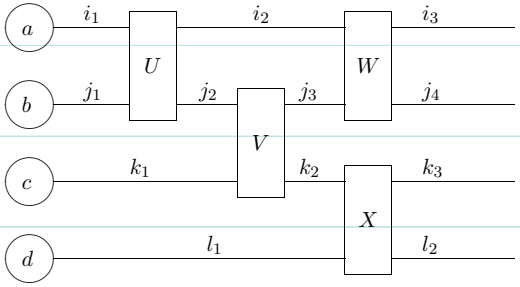
$$c_{i_1 \ldots i_n} = \alpha_{i_1} \beta_{i_2} \gamma_{i_3} \ldots \delta_{i_n}$$

then $\alpha'_{i_1} \beta'_{i_2} \ldots \delta'_{i_n} = \left( V^{j_1 j_2}_{i_1 i_2} \alpha_{j_1} \beta_{j_2} \right) \left( \gamma_{i_3} \ldots \delta_{i_n} \right)$

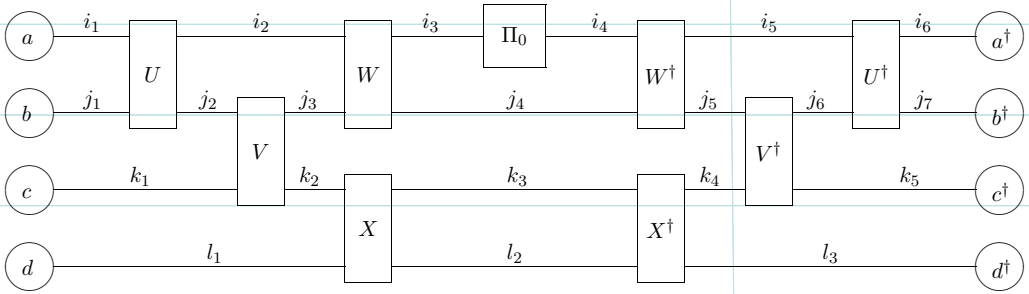
update can be computed in poly(n) time.

Hence presence of (increasing multiparty) entanglement is necessary for q. comp. benefit but it's not sufficient!

More sophisticated tensor contraction schemes eg.—




$$= a^{i_1} b^{j_1} U_{i_1 j_1}^{i_2 j_2} c^{k_1} V_{j_2 k_1}^{j_3 k_2} W_{i_2 j_3}^{i_3 j_4} d^{l_1} X_{k_2 l_1}^{k_3 l_3}$$




$= \text{prob}(0)$

$\Pi_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = |0\rangle\langle 0|$ on 1st qubit

$$\text{prob}(0) = \langle \psi_f | \Pi_0 \otimes I \cdots \otimes I | \psi_f \rangle$$

# Theory of tensor network contractions

Want to choose order of contractions so that intermediate tensors do not accumulate too many indices!

$k$ indices $\sim 2^k$ components so try to keep $k = O(\log n)$

- Optimal contraction ordering $\sim$ tree width / tree decomposition of graph associated to circuit (gates = vertices, lines = edges)
  NP hard problem generally

Application to quantum circuits:

Markov & Shi quant-ph/0511069 (2005)

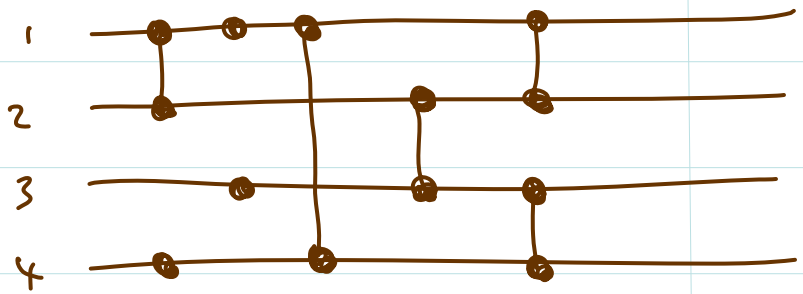
Example (R. Jozsa quant-ph/0603163 (2006))

C any poly sized circuit of 1 & 2 qubit gates on n qubits

- Input any product state
- Output - measure 1st line
- For each qubit line $i$ let $D_i$ = number of 2-qubit gates that touch or cross line $i$
- Let $D = \max D_i$

Then output can be classically simulated in time poly$(2^D)$

e.g.




⋮ denote gate actions

$D_1 = 3$

$D_2 = 4$ so $D = 4$

$D_3 = 3$

$D_4 = 2$

Thus any log-depth circuit of bounded range gates can be classically efficiently simulated.

[log depth + bded range $\Rightarrow D = O(\log n)$. Contract all line 1 indices ($i$'s) first, then all line 2 ($j$'s) etc gives explicit contraction ordering. ]

Tensor contraction techniques apply to any tensors whatever!
Get further (ingenious!) simulation results for special kinds of circuits – comprising only restricted kinds of gates
Recall - universal set of gates only a smallish special set ....

Will discuss 2 examples:

Clifford circuits (Gottesman-Knill theorem)

Matchgate circuits (Valiant's theorem)

# Clifford Operations

Recall Pauli operations $\pm i, I, X, Y, Z$. on 1-qubit.

Pauli group on $n$ qubits

$$P_n = \{ P_1 \otimes \cdots \otimes P_n : P_i\text{'s are all 1-qubit Pauli's} \}$$

$P_n$ subgroup of $U(2^n)$.

Clifford operation $C$ on $n$ qubits

$$C(P_1 \otimes \cdots \otimes P_n) C^\dagger = \tilde{P}_1 \otimes \cdots \otimes \tilde{P}_n$$

maps any Pauli product to a Pauli product under conjugation.

Clifford group $C_n$ on $n$-qubits

is <u>normaliser</u> of $P_n$ in $U(2^n)$

**Theorem:** $C$ is Clifford iff

$C$ is circuit of $H$, $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ & CNOT gates.

(so have all Pauli's, SWAP etc too)

**Theorem** (Gottesman-Knill variant)

Consider any polysized circuit of basic Clifford gates with (1) input is any product state

(2) output is measurement on 1st line

Then the output may be classically efficiently simulated.

↑ can generate entanglements

↖ contraction orderings don't work!

eg. $|0\rangle|0\rangle \xrightarrow{H_1} \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle$

$\xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$ etc.

# Proof outline

Pauli $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| = \Pi_0 - \Pi_1$

$C$ any Clifford circuit on $n$ qubits (comprising $H, P, \text{CNOT}$ gates)

Input $|\psi_0\rangle = |a_1\rangle .. |a_n\rangle$ Output probs $p_i = \langle \psi_f | \Pi_i | \psi_f \rangle$

$|\psi_f\rangle = C|\psi_0\rangle$ $\langle \psi_f | = \langle \psi_0 | C^\dagger$

Write $Z_1 = Z \otimes I \otimes .. \otimes I$

So $p_0 - p_1 = \langle \psi_0 | C^\dagger Z_1 C | \psi_0 \rangle$

$= \langle \psi_0 | P_1 \otimes .. \otimes P_n | \psi_0 \rangle$ ← since $Z_1 \in \mathcal{P}_n$ and $C$ Clifford

$= \prod_{i=1}^{n} \langle a_i | P_i | a_i \rangle$

Can compute in $O(n)$ time: product of $n$ $2 \times 2$ matrix computations.

Also: update rule for Pauli products by Clifford conjugations is easy to compute classically.

# Valiant's theorem (classical simulation of unitary matchgate circuits)

Consider 2-qubitgates

Matchgates

$$G(A,B)=\begin{bmatrix} p & 0 & 0 & q \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ r & 0 & 0 & s \end{bmatrix} \qquad A=\begin{bmatrix} p & q \\ r & s \end{bmatrix} \qquad B=\begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

with $A, B$ both in $su(2)$ (or $U(2)$ with same determinant)

i.e. $A$ acts in 00/11 and $B$ acts in 01/10 subspaces.

Consider any (polysized) circuit of $G(A,B)$ gates with:

- $G(A,B)$ acts on nearest-neighbour (nn) lines only
- input is any product state
- final measurement is in $Z_k$ basis on any single line $k$

Then output can be classically efficiently simulated.

More precisely: can compute $\langle \psi_{out} | Z_k | \psi_{out} \rangle = P_0 - P_1$ to $k$ digits in poly$(n,k)$ time. □

Warning: non-n.n. use of $G(A,B)$ gates not allowed !!

SWAP $= G(I,X)$ not included (fails by just a minus sign--)

But $S = G(Z,X) = G(Z,I)\,G(I,X) =$ CZ. SWAP is allowed.

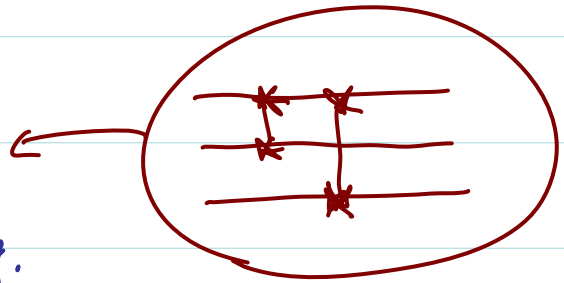
Theorem:

(a) n.n. $G(A,B)$ gates with SWAP is universal for quantum computing.

More strongly (limited use of SWAP suffices):

(b) (R. Jozsa & A. Miyake 2008)

$G(A,B)$ gates acting on n.n. and next-n.n. is efficiently universal for quantum computing.



# Proof ingredients for Valiant's theorem

For $n$ qubit lines

Introduce $2n$ operators $c_1, \ldots, c_{2n}$ with

$$\{c_\mu, c_\nu\} \equiv c_\mu c_\nu + c_\nu c_\mu = 2\delta_{\mu\nu} I$$

← Clifford algebra $C_{2n}$ on $2n$ generators

General vector in this algebra is

$$\sum_{i_1 < \ldots < i_k \text{ maybe empty}} a_{i_1 \ldots i_k} c_{i_1} \ldots c_{i_k}$$

← $c^2 = I$ and $c$'s anticommute.

Hence $2^{2n} = (2^n)^2$ dimensional as a vector space

We'll be interested in matrix representations

$\sim 2^n \times 2^n$ sized (Hermitian) matrices.

# Some Further Pure Algebra

Quadratic Hamiltonian $H = i \sum_{\mu \neq \nu = 1}^{2n} h_{\mu\nu} c_\mu c_\nu$

$H$ hermitian & $c_\mu c_\nu = -c_\nu c_\mu \Rightarrow h_{\mu\nu}$ real antisymmetric $2n \times 2n$ matrix.

Introduce Gaussian gate $U = e^{iH}$ ($H$ quadratic) ← exponential ≡ powerseries

**Basic Theorem** If $U$ is Gaussian then

$$U^\dagger c_\mu U = \sum_{\nu=1}^{2n} R_{\mu\nu} c_\nu$$

with $R \in SO(2n, \mathbb{R})$ [infact $R = e^{4h}$]

Significance: $U^\dagger c_\mu U$ could finish up anywhere in exponentially big linear space $\mathcal{C}_{2n}$ (cf $e^{iH} \sim$ powers of all $c_\nu$ products etc.)

but happens to always stay in $2n$ (poly-many) dimensions!

**Proof idea:**

Consider $U(t) = e^{iHt}$ !

$$c_\mu(t) = U(t)\, c_\mu\, U(t)^\dagger$$

Then $$\frac{dc_\mu}{dt} = [H, c_\mu]$$

But: $[c_\mu, c_{\nu_1} c_{\nu_2}] = c_\mu c_{\nu_1} c_{\nu_2} - c_{\nu_1} c_{\nu_2} c_\mu = 0$ if $\mu \neq \nu_1, \nu_2$

$[c_\mu, c_\mu c_\nu] = \;\cdots\; = 4 c_\nu$

So $$\frac{dc_\mu}{dt} = 4 h_{\mu\nu} c_\nu$$ and result follows immediately at $t = 1$

(recalling: infinitesimal rotations $\equiv$ antisym matrices) //

# Intuitive analogy with Clifford circuit formalism

| | | |
|---|---|---|
| Clifford algebra $\mathcal{C}_{2n}$ | Unitary group $U(2^n)$ | "expon. big" |
| Linear part of $\mathcal{C}_{2n}$ | Pauli group $\mathcal{P}_n$ | "only polynomially small" |
| Gaussian operations | Clifford operations | "preserves small part by conjugation" |

Next: introduce explicit matrix representation of $\mathcal{C}_{2n}$ for which matchgates appear as Gaussian gates.

# Jordan-Wigner representation of $C_{2n}$ on $n$ qubits

$c_\mu$'s are Hermitian, Product operators and Pauli's too.

$c_1 = X\,I\,..\,I$

$c_2 = Y\,I\,..\,I$

$c_3 = Z\,X\,I\,..\,I$

$c_4 = Z\,Y\,I\,..\,I$

$c_{2k-1} = Z\,...\,Z\,X\,I\,..\,I$

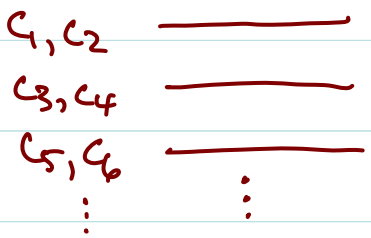
$c_{2k} = Z\,..\,Z\,Y\,I\,..\,I$

($k^{th}$ slot)

Easy to check $c_\mu c_\nu + c_\nu c_\mu = 2\delta_{\mu\nu} I$

Also $Z_k = i\, c_{2k-1}\, c_{2k}$

# Gaussian gates in JW representation




First look at qubit lines 1 & 2

Associated quadratic terms from $c_1, c_2, c_3, c_4$

| | |
|---|---|
| $ic_1c_2$ | $ZI$ |
| $ic_1c_3$ | $YX$ |
| $ic_1c_4$ | $YY$ |
| $ic_2c_3$ | $XX$ |
| $ic_2c_4$ | $XY$ |
| $ic_3c_4$ | $IZ$ |

- all preserve 00/11 & 01/10 subspaces
- all trace free, 6 parameters worth

Hence get $su(2) \oplus su(2)$ in subspaces

i.e. get precisely $G(A,B)$'s on lines 1 & 2.

Similarly: get all <u>n.n.</u> $G(A,B)$'s as Gaussian gates.

Note:

① next-n.n. (& more distant) $G(A,B)$'s not Gaussian

eg. replace $ic_1c_3 = YX$ by $ic_1c_5$, get $YZX$ $(\neq YIX)$

lines 1 & 2 lines 1 & 3 from JW formulae.

② Also have more general quadratic H's using more c's associated to distant lines i.e. more Gaussian gates than just n.n. $G(A,B)$'s. But:

Theorem (RTdAM) If $H = i\sum h_{\mu\nu} c_\mu c_\nu$ is general quadratic hamiltonian then the Gaussian gate $U = e^{iH}$ on n qubit lines is always expressible as a circuit of n.n. $G(A,B)$ gates ($O(n^3)$ of them).

i.e. in context of polysized circuits get nothing new beyond n.n. $G(A,B)$'s.

Now: back to simulating n.n. matchgate circuits

Let $M = M_1 M_2 .. M_{poly(n)}$ be any polysized n.n. matchgate circuit

Consider $|\psi_{in}\rangle = |0..0\rangle$, final $Z_1$ measurement.

(General $|\psi_{in}\rangle = |a_1\rangle .. |a_n\rangle$, final $Z_k$ similar)

Then have $p_0 - p_1 = \langle \psi_{in} | M^\dagger Z_1 M | \psi_{in} \rangle$ ← want to compute this efficiently

n.n. matchgates are Gaussian so let:

$\tilde{R}$ be associated $SO(2n)$ matrix obtained by multiplying rotations of (poly-many) gates in $M = M_1 M_2 .. M_{poly(n)}$

So $M^\dagger c_i M = \tilde{R}_{ij} c_j$ and recall $Z_1 = -i c_1 c_2$

Hence

$$M^\dagger Z_1 M = -i (M^\dagger c_1 M)(M^\dagger c_2 M) = -i \left( \sum_j \tilde{R}_{1j} c_j \right) \left( \sum_k \tilde{R}_{2k} c_k \right)$$

and

$$p_0 - p_1 = -i \sum_{j,k=1}^{2n} \tilde{R}_{1j} \tilde{R}_{2k} \langle 0..0 | c_j c_k | 0..0 \rangle$$

Claim:

$$p_0 - p_1 = -i \sum_{j,k=1}^{2n} \tilde{R}_{1j} \tilde{R}_{2k} \langle 0\ldots0 | c_j c_k | 0\ldots0 \rangle$$

is computable in poly(n) time.

- $\tilde{R}$ is product of poly-many $2n \times 2n$-sized matrices ✓
- $c_i$'s are product (Pauli) operators, hence so is

$$c_j c_k = \prod_{i=1}^{n} P_i \text{ say.}$$

So

$$\langle 0\ldots0 | c_j c_k | 0\ldots0 \rangle = \prod_{i=1}^{n} \langle 0 | P_i | 0 \rangle$$

is poly-time computable.

Hence: n.n. matchgate circuits are classically efficiently simulatable. □

## Further complexity issues

Clifford and matchgate circuits can be classically efficiently simulated but their computational power appears to be strictly weaker than full classical poly-time computation!

### Clifford circuits

Update rule for $C^\dagger (P_1 \otimes .. \otimes P_n) C \rightsquigarrow (P_1' \otimes .. \otimes P_n')$

to compute evolution of $Z \otimes I \otimes . \otimes I = Z_1$

is classical poly-time, but more - need only to compute bit sums

$i_1 .. i_k \longrightarrow i_1 \oplus . \oplus i_k$ so need only CNOT / NOT gates.

**Theorem** (Aaronson, Gottesman): Computational power of (log-space uniform) families of Clifford circuits coincides with the classical complexity class called $\oplus L$.

# Matchgate circuits

Simulation needs mult$^n$ of poly(n) matrices of size $O(n) \times O(n)$

It's classical poly-time but more—

can do much in parallel!

- Mult of two $O(n)$-sized matrices can be done in $O(\log n)$ depth
- Mult of poly(n) matrices can be done with $O(\log n)$-depth circuit of pairwise matrix mults.

So : $O((\log n)^2)$ - depth in all.

So matchgate circuit power $\subseteq NC^2 \subseteq P$.

Can say still more — (R.J, B. Kraus, A. Miyake, J. Watrous, arXiv:0809.1467 (2009))

The $O(n)$-sized matrices are rotations so can be viewed as unitary operations on $O(\log n)$ qubits.

Then can show: matchgate circuits on n qubit lines

$\Updownarrow$ equivalence ("exponential width compression")

general universal quantum computation on $O(\log n)$ qubit lines.

# A final intriguing observation:

Let $\mathcal{C}$ be any class of "nontrivial" quantum circuits that are cl.-simulatable.

If "computational power" of $\mathcal{C}$ is full classical poly-time

↑ i.e. "can make any classical gate by a circuit in $\mathcal{C}$"

Then quantum computation = classical computation!

Introduce Toffoli gate: classical 3-bit C-C-NOT $(i)(j)(k) \rightsquigarrow (i)(j)(k \oplus ij)$

(universal for classical computation)

Theorem (Y. Shi) If $Q$ is any 1-qubit gate that's basis-changing

i.e. $\{|0\rangle, |1\rangle\}$ not mapped to $\{e^{i\alpha}|0\rangle, e^{i\beta}|1\rangle\}$

then $\{Q, \text{Toffoli}\}$ is universal for quantum computation. □