# An overview of quantum algorithms

Ashley Montanaro

School of Mathematics, University of Bristol
Phasecraft Ltd

24 April 2024

# Quantum computers

Quantum computers are designed to do things that classical computers cannot. But to achieve a quantum speedup requires a quantum algorithm.

Most quantum algorithms can be divided into 5 categories:

| Algorithm | Speedup | Example |
|---|---|---|
| Simulation of quantum systems | Exponential | Lloyd |
| Breaking cryptographic codes | Exponential | Shor |
| Optimization / combinatorial search | Square-root | Grover |
| High-dimensional linear algebra | Exponential? | HHL |
| Quantum heuristics | Unknown | QAOA |

The Quantum Algorithm Zoo lists hundreds of papers on quantum algorithms.

# Near-term vs. long-term quantum algorithms

Our field often separates quantum computing into the NISQ era ("Noisy Intermediate-Scale Quantum", ie. no error-correction) and the fault-tolerant era [Preskill 1801.00862].

But we can instead organise by number of instructions enabled [Bacon '24]:

| **Era** | **Gates** | **Example** |
|---|---|---|
| KISQ | $10^3$ | Quantum supremacy [Morvan et al 2304.11119] |
| MISQ | $10^6$ | Early cond-mat / materials [Cade et al 1912.06007, Clinton et al 2205.15256] |
| GISQ | $10^9$ | Quantum chemistry [Lee et al 2011.03494] |
| TISQ | $10^{12}$ | Factoring [Gidney and Ekera 1905.09749] |

(ISQ = "–Instruction Scale Quantum")

No quantum algorithms are inherently NISQ; but some quantum algorithms inherently need fault-tolerance.

# Quantum simulation

The most important early application of quantum computers is likely to be quantum simulation: modelling a quantum-mechanical system on a quantum computer.

Applications include quantum chemistry, superconductivity, novel materials, high-energy physics, ... [Georgescu et al 1308.6253]

Quantum systems are represented by Hamiltonians: exponentially big matrices $H$ represented in an efficient way, e.g. the Heisenberg model

$$H = \sum_{\langle i,j \rangle} X_i X_j + Y_i Y_j + Z_i Z_j.$$

- Static simulation: e.g. compute ground energy $\lambda_{\min}(H) \Rightarrow$ QMA-complete.
- Time-dynamics simulation: e.g. compute $\langle 0|e^{-iHt}|0\rangle \Rightarrow$ BQP-complete.

# Quantum simulation algorithms: Ground states

Although producing ground states is expected to be hard in the worst case, many approaches have been developed which may work well in practice, e.g.:

- The Variational Quantum Eigensolver (VQE) [Peruzzo et al 1304.3061]. Optimize over a family of quantum circuits ("ansatz") to minimize the energy.
- Quantum imaginary-time evolution (QITE) [Motta et al 1901.07653]. Approximate the operator $e^{-\Delta H}$.
- Adiabatic evolution [Farhi et al quant-ph/0001106]. Slowly change an "easy" Hamiltonian into a "hard" one, maintaining the ground state.
- The Dissipative Quantum Eigensolver [Cubitt 2303.11962]. Perform weak measurements to gradually project onto the ground state.

Why do we care about producing ground states? We can directly extract useful information from them. For example, voltage profiles of batteries.

# Quantum simulation algorithms: Time-dynamics

Simulating time-dynamics is efficient in principle, but not always efficient in practice!

We want to implement the unitary operator $e^{-iHt}$ for (e.g.) a $k$-local Hamiltonian $H = \sum_j H_j$.

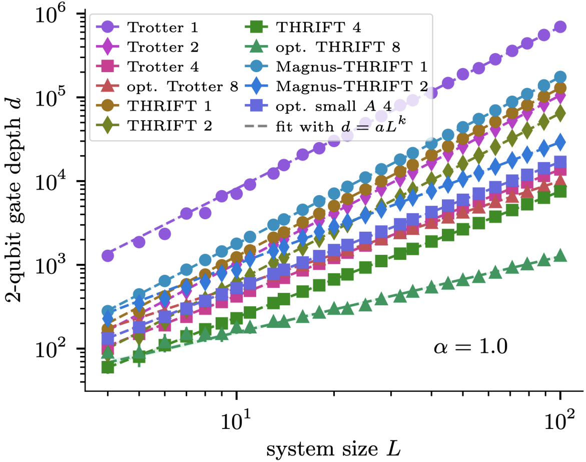
We can do this by, e.g.:

- Product (Trotter) formulae, e.g. $e^{-iHt} \approx (e^{-iH_1 t/\Delta} e^{-iH_2 t/\Delta} \dots)^{\Delta}$ [Lloyd '96]
- Taylor series (LCU) methods, e.g. $e^{-iHt} \approx \sum_{j<J} \frac{(-itH)^j}{j!}$ [Berry et al 1412.4687]
- Quantum signal processing [Low and Chuang 1606.02685]

Many other techniques are known!

# Quantum simulation algorithms: Time-dynamics

Consider the $1 \times L$ Ising model with transverse field, $H = \sum_j Z_j Z_{j+1} + \sum_k X_k$:




Quantum circuit depths to evolve for time $L$ with error 0.01 [Bosse et al 2403.08729]

# Cryptography

Many (though not all) cryptosystems are known to be vulnerable to quantum attack.

| **Cryptosystem** | **Problem** | **Quantum algorithm** |
|---|---|---|
| RSA | Factoring | Shor |
| Elliptic curve | Discrete log | Shor |
| Lattice | Dihedral HSP (?) | Watch this space... |
| McEliece | Error-correction | ? |

The field of post-quantum cryptography aims to develop cryptosystems that are secure against quantum attack. NIST standardisation process has been running since 2016!

See e.g. [Gidney+Ekera 1905.09749] for a detailed analysis, showing that a 2048-digit integer can be factorised in 8 hours with 23 million physical qubits.

# Search and optimization

One of the most basic problems in computer science is unstructured search.

- Imagine we have access to a function $f : \{0, 1\}^n \to \{0, 1\}$ which we treat as a black box.

- We want to find an $x$ such that $f(x) = 1$.

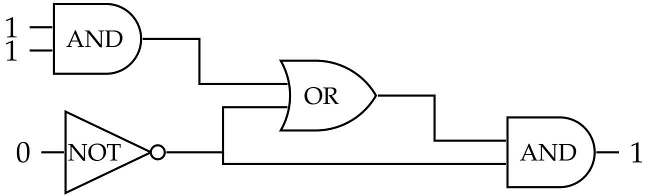
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

- On a classical computer, this task could require $2^n$ queries to $f$ in the worst case. But on a quantum computer, Grover's algorithm [Grover quant-ph/9605043] can solve the problem with $O(\sqrt{2^n})$ queries to $f$ (and bounded failure probability).

# Applications of Grover's algorithm

Grover's algorithm gives a speedup over naïve algorithms for any decision problem in the complexity class NP, i.e. where we can verify the solution efficiently.

- For example, in the Circuit SAT problem we would like to find an input to a circuit on $n$ bits such that the output is 1:




- Grover's algorithm improves the runtime from $O(2^n)$ to $O(2^{n/2})$: applications to design automation, circuit equivalence, model checking, ...

# Beyond Grover's algorithm

Grover's algorithm accelerates classical unstructured search.

We can also accelerate other classical subroutines quadratically:

- Backtracking and branch-and-bound [AM 1509.02374, AM 1906.10375]
- Dynamic programming [Ambainis et al 1807.05209]
- Random walks [Szegedy '04]

These then have many applications, e.g.:

- Quantum speedup of the Travelling Salesman Problem on bounded-degree graphs [Moylett et al 1612.06203]
- Finding shortest vectors in lattices for cryptographic applications [Alkim et al. '15, del Pino et al. '16]
- ...

# Challenges associated with these algorithms

Quantum algorithms based on Grover's algorithm, quantum walks and similar techniques tend to have the following features:

- They achieve an at most quadratic speedup over an analogous classical algorithm;
- They require deep quantum circuits (and hence fault-tolerance).

Putting these two together, they face a significant challenge from error-correction overheads:

- Graph colouring / boolean satisfiability: speedup factor of $\sim 10^5$ (ignoring cost of fault-tolerance processing) but $\sim 10^{12}$ physical qubits required [Campbell et al 1810.05582] (see [Babbush et al 2011.04149] for an even more pessimistic outlook)

Can we solve optimization problems using quantum computers in the near term?

# The quantum approximate optimization algorithm

We can apply the VQE framework to solve classical optimization problems by setting

$$H = \sum_{x \in \{0,1\}^n} C(x) |x\rangle\langle x|$$

where $C(x)$ is a cost function. The ground state of $H$ is then the lowest-cost $x$.

[Farhi et al 1411.4028] proposed the following variational method:

- Start with $|+^n\rangle$
- Apply $e^{i\gamma H}$
- Apply $e^{i\beta \sum_j X_j}$

Repeat steps 2 and 3 (with different parameters) $p$ times. Then optimize over the parameters $\beta_1, \ldots, \beta_p, \gamma_1, \ldots, \gamma_p$.

An essentially identical algorithm was described by [Hogg '00].

# Performance of the quantum approximate optimization algorithm

Precisely how well QAOA performs on a given problem is generally hard to determine.

- It's known that QAOA performs well with many layers (can reproduce the performance of Grover's algorithm [Jiang et al 1702.02577]) and is hard to simulate classically even with 1 layer [Farhi and Harrow 1602.07674].
- Its performance on optimization problems can be analysed but generally the bounds don't outperform classical methods.
- QAOA can be applied to constraint satisfaction problems (e.g. random $k$-SAT) and its performance analysed [Boulebnane and AM 2208.06909] – scaling seems better than the best classical algorithms considered.

# “Solving” linear equations

A basic task in mathematics and engineering:

**Solving linear equations**

Given access to a $d$-sparse $N \times N$ matrix $A$, and $b \in \mathbb{R}^N$, output $x$ such that $Ax = b$.

One “quantum” way of thinking about the problem:

**“Solving” linear equations**

Given the ability to produce the quantum state $|b\rangle = \sum_{i=1}^N b_i|i\rangle$, and access to $A$ as above, produce the state $|x\rangle = \sum_{i=1}^N x_i|i\rangle$.

Theorem: If $A$ has condition number $\kappa$ $(= \|A^{-1}\|\|A\|)$, $|x\rangle$ can be approximately produced in time $\mathrm{poly}(\log N, d, \kappa)$ [Harrow et al 0811.3171]

# Notes on this algorithm

The algorithm (approximately) produces a state $|x\rangle$ such that we can extract some information from $|x\rangle$. Is this useful?

- We could use this to e.g. determine whether two sets of linear equations have (approximately) the same solution – not clear how to do this classically.
- Achieving a similar runtime classically would imply that all quantum computations could be simulated!

Some applications of this algorithm include:

- Electromagnetic scattering cross-sections using the finite element method [Clader et al 1301.2340] [AM+Pallister 1512.05903]
- "Solving" differential equations [Leyton+Osborne 0812.4423] [Berry 1010.2745]
- Recommendation systems and other problems in machine learning (e.g. [Kerenidis+Prakash 1603.08675]) – but note "quantum-inspired" competition [Tang 1807.04271]!

# What are the minimal gate counts to do something useful?

| Task | 2-qubit gates |
|---|---|
| One layer of VQE/Trotter for $10 \times 10$ Fermi-Hubbard | 2,000 |
| One layer of VQE for $SrVO_3$ [Clinton et al 2205.15256] | 7,500 |
| One layer of VQE/Trotter for Kagome Heisenberg on 100 qubits | 350 |
| TDS for $1 \times 100$ Ising model with weak transverse field for time 100 | $\approx 50{,}000$ |
| TDS for $5 \times 5$ Fermi-Hubbard for time 7 | $\approx 75{,}000$ |

Some benchmark experiments:

| Hardware | Experiment | 2-qubit gates |
|---|---|---|
| IBM | Kicked Ising dynamics | 2,880 |
| Google | Fermi-Hubbard TDS | 608 |
| Google | "Quantum supremacy" 2023 | 880 |
| Quantinuum | Holographic quantum dynamics | 2,130 |

# Conclusions

There are many quantum algorithms, solving many different problems, some of which achieve substantial speedups over their classical counterparts.

Important future research directions include:

- Finding more practical applications for these algorithms;
- Analysing their complexity in detail;
- New ideas for quantum algorithm design;
- Getting the most out of near-term quantum computers.

Further reading:

- Quantum algorithms: an overview [AM, 1511.04206]
- Quantum algorithm design: techniques and applications [Shao et al, Journal of Systems Science and Complexity, 2019]
- Noisy intermediate-scale quantum (NISQ) algorithms [Bharti et al, 2101.08448]
- Tutorial talk by Andrew Childs at QIP 2021
  `https://www.cs.umd.edu/~amchilds/talks/qip21.pdf`
- Tutorial talk by Andrew Childs on quantum simulation
  `https://www.cs.umd.edu/~amchilds/talks/sim.pdf`

Advert: We are hiring at Phasecraft!

Thanks!