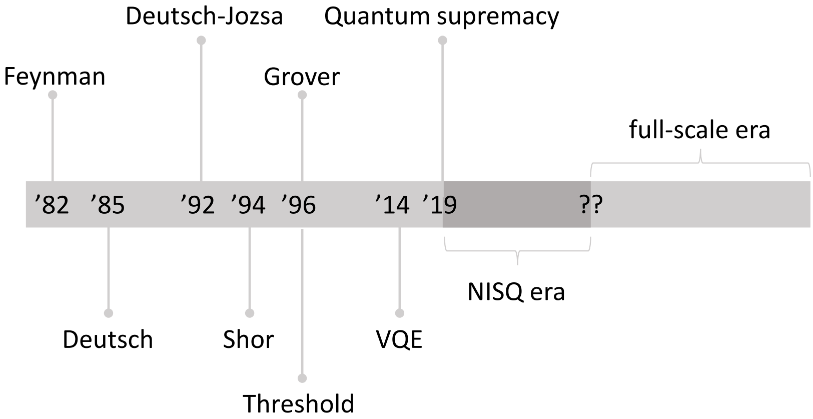
Quantum Computing (CST Part II)

Lecture 16: Case Studies in Near-term Quantum Computation

In less than ten years quantum computers will begin to outperform everyday computers, leading to breakthroughs in artificial intelligence, the discovery of new pharmaceuticals and beyond.
Jeremy O'Brien (2016)

Timeline of quantum computing



NISQs and full-scale fault-tolerant quantum computers

The successful quantum supremacy experiment, demonstrated by Google in 2019, has heralded the start of the *NISQ era*. "NISQ" stands for *noisy intermediate-scale quantum* (computer), a name coined by John Preskill.

The NISQ era is an exciting time, as experimentalists begin to use small-scale quantum hardware to gain better understanding of, for example, how to encode chemical properties into qubits. It is, however, becoming increasingly clear that the variational algorithms available in the NISQ era are unlikely to scale to give an actual quantum advantage.

The quantum algorithms we have studied in this course typically require full-scale fault-tolerant quantum computers, and the time when such technology exists I have termed the *full-scale era* in this lecture.

Quantum algorithms not covered in the course

Some of the most important quantum algorithms that we haven't had chance to study in the course are:

- Quantum counting, quantum amplitude estimation (QAE) & quantum Monte Carlo integration (QMCI);
- HHL;
- The quantum singular-value decomposition (QSVD).

Quantum counting, QAE & QMCI

Approximate quantum counting can be used to estimate the number of marked elements, and hence the number of Grover iterates required in unstructured search.

The same essential approach can be used to estimation the *amplitude*, $a = \sin^2 \theta$ of any quantum state:

$$|\psi\rangle = \cos\theta\, |\Phi_0\rangle\, |0\rangle + \sin\theta\, |\Phi_1\rangle\, |1\rangle$$

It turns out that this can be used as a sub-routine to achieve a quadratic advantage in quantum Monte Carlo integration – which in turn speeds up estimation and forecasting in a wide-variety of applications.

HHL

HHL is a quantum algorithm invented in 2008 for approximately solving sparse systems of linear equations, known by the initials of its inventors, Aram Harrow, Avinatan Hassidim and Seth Lloyd. From the HHL Wikipedia article:

"Due to the prevalence of linear systems in virtually all areas of science and engineering, the quantum algorithm for linear systems of equations has the potential for widespread applicability."

HHL has the interesting property that it combines many of the fundamental quantum algorithms that we have studied:

- Quantum phase estimation
- Hamiltonian (quantum) simulation
- Amplitude amplification (a generalisation of Grover's algorithm)

The quantum singular-value transform

The quantum singular value transform is an extremely general framework, dubbed the grand unifier of quantum algorithms as each of:

- matrix inversion;
- unstructured search;
- factoring;
- Hamiltonian simulation,

can be cast as instances thereof.

Overview of quantum algorithms

Quantum algorithm zoo lists over 60 quantum algorithms, some of the main ones are:

Algorithm	**Function**	**Speed-up**	**Era**
Shor	factoring	super-polynomial	full-scale
Grover	search	polynomial	full-scale
HHL	linear algebra	super-polynomial	full-scale
QPE	chemistry	super-polynomial	full-scale
QMCI	estimation	polynomial	full-scale
VQE	chemistry	heuristic	NISQ
Annealing	optimisation	heuristic	NISQ
QAOA	optimisation	heuristic	NISQ

Overview of quantum algorithms

Quantum algorithm zoo lists over 60 quantum algorithms, some of the main ones are:

Algorithm	**Function**	**Speed-up**	**Era**
Shor	factoring	super-polynomial	full-scale
Grover	search	polynomial	full-scale
HHL	linear algebra	super-polynomial	full-scale
QPE	chemistry	super-polynomial	full-scale
QMCI	estimation	polynomial	full-scale
VQE	chemistry	heuristic	NISQ
Annealing	optimisation	heuristic	NISQ
QAOA	optimisation	heuristic	NISQ
Quantum machine learning		various	both?

Quantum machine learning

“Quantum machine learning” is a buzz-word heavy slide title, but what does it actually mean? Crudely, it can be divided into three categories:

		Algorithm	
		quantum	classical
Data	quantum		
	classical		▨

Quantum machine learning (cont.)

- **Quantum machine learning on classical data** refers to the use of quantum algorithms to enhance the performance of conventional machine learning algorithms. For example, quantum optimisation (annealing or QAOA), search (Grover) and / or linear system solving (HHL) may be called as subroutines by some otherwise classical machine learning algorithm.
- **Classical machine learning on quantum data** refers to the use of conventional, classical learning techniques to learn something about some quantum data. Quantum state tomography is a basic example.
- **Quantum machine learning on quantum data** is, it could be argued, true quantum machine learning, in the sense that we want to discern some information from a quantum data-set, which may not be possible if that quantum data were simply measured and classical learning applied.

In short, the second and third items differ because, in the former the quantum state is measured and thus collapsed into classical data on which classical machine learning is applied, whereas in the latter quantum operations are applied to the quantum data.

Areas quantum computing is expected to impact

In chronological order (from most imminent to most distant), quantum computing is expected to have a big impact on the following:

- **Chemistry**: the expectation of exponential quantum advantage in chemical simulations is already exciting people working in drug discovery, oil and gas and many others, as well as myriad applications in materials science.
- **Optimisation, estimation and QML**: the hope of quadratic (and potentially exponential) quantum advantage could potentially touch virtually all areas of engineering and operations research.
- **Security**: as has long been anticipated, when we have full-scale quantum computers, Shor's algorithm will be game-changer for computer security. But so too, perhaps, will be QKD.

Note that this list is by no-means exhaustive, but is simply here to give a flavour of an optimistic view of what things are to come.

How many qubits do we need?

Asking how many qubits we need to do something than cannot be achieved classically is a bit like asking how long is a piece of string. Nevertheless, to give a rough idea:

- **Quantum supremacy** (a mathematically well-defined but useless sampling problem): 53 qubits[1].
- **Quantum chemistry** (simulation of Caffeine): 160 logical qubits[2].
- **Estimation** (e.g. QMCI in finance): 100 – 1000 logical qubits[3].
- **Factoring**: (breaking RSA-2048): 20M physical qubits[4].

[1] https://www.nature.com/articles/s41586-019-1666-5

[2] e.g. https://www.ft.com/content/154a1cf4-ad07-11e8-94bd-cba20d67390c

[3] https://doi.org/10.1017/dce.2022.36

[4] https://cacm.acm.org/news/237303-how-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/fulltext

From the NISQ era to the full-scale era

Fault-tolerance is the feature that distinguishes the full-scale era from the NISQ era, and this will require an error correction overhead estimated to be in the region 20–1000. That is, it will take 20–1000 physical qubits to make each "clean" logical qubit.

Qubit fidelities and error correcting codes may well improve, bringing this number down, but the fact remains that a serious scaling-up of the number of qubits in a quantum computer is needed to build a fault-tolerant quantum computer.

This in turn has led some in the quantum computing community to talk about the need for a "quantum transistor" – a highly scalable physical realisation for a qubit, that changes the game for quantum computing in the same way that the transistor did for classical computing.

Types of qubit

There are various proposals for physically realising a qubit, of which the most promising are superconducting qubits and trapped-ion qubits.

- At present, superconducting quantum computers have the most qubits, and superconducting qubits offer fast gate times.
- On the other hand, trapped-ion qubits have the highest fidelity and longest coherence times, and essentially have all-to-all connectivity, unlike the restricted (planar) connectivity of most superconducting quantum computers.
- Other technologies include: silicon qubits; nitrogen-vacancy qubits; and optical (photonic) qubits.

How good is a particular quantum computer?

The total number of qubits tends to grab the headlines, but how good a particular quantum computer is actually depends on three factors:

1. The number of qubits.
2. The quality of those qubits (fidelity).
3. The connectivity (what overhead will be incurred to move the qubits around such that they can interact).

Quantum volume is a measure that has been proposed to quantify how good a given quantum computer is, incorporating these three factors. The quantum volume of a quantum computer is given by:

$$Q_v = 2^{(\min(n,d))}$$

where n is the number of qubits and d is the depth of a random circuit that can be executed before an error is expected to occur.

Quantum volume

- The depth term, d, is a function of both the fidelity and the depth overhead incurred when executing a circuit consisting of random 2-qubit interactions.
- Therefore the quantum volume is increased for quantum computers with higher connectivity, as fewer SWAP gates will be needed to rearrange interacting qubits to be local, and so the depth overhead will be smaller.
- The presence of the $\min$ term in the definition indicates whether the performance of a given quantum computer is limited by a lack of qubits or poor fidelity / connectivity of the qubits – i.e., a more nuanced picture than simply quoting the number of qubits.
- Quantum volume has been conceptualised so that it gives a reasonable benchmark of the general performance of near-term quantum computers.
- However some researchers refute that random circuits are appropriate for this, and instead assert that it part of the role of quantum software design to execute algorithms in an efficient manner, given the physical locality constraints of the hardware.

The state of the art in quantum hardware

Three players lead the way in quantum hardware:

Company	**Headline claim**	**Qubits**	**Quantum Volume**
Google	Quantum supremacy	72	–
IBM	Most qubits	433	$256 = 2^8$
Quantinuum	Highest quantum volume	20	$32768 = 2^{15}$

Quantum-inspired classical algorithms

One of the most important problems in data-science is the construction of *recommendation systems*. Suppose that we have n products and a purchase history of m users, from which we need to make product recommendations:

- Until 2016 only techniques which run in time $\mathcal{O}(\text{poly}(mn))$ were known.
- In 2016 Iordanis Kerenidis and Anupam Prakash published a quantum algorithm to achieve this task in time $\mathcal{O}(\text{poly}\log(mn))$. That is, an exponential speed-up.
- Then in 2018 Ewin Tang published a classical algorithm inspired by Kerenidis and Prakash's quantum algorithm that also achieves the task in time $\mathcal{O}(\text{poly}\log(mn))$.

Quantum-inspired classical algorithms

One of the most important problems in data-science is the construction of *recommendation systems*. Suppose that we have n products and a purchase history of m users, from which we need to make product recommendations:

- Until 2016 only techniques which run in time $\mathcal{O}(\text{poly}(mn))$ were known.
- In 2016 Iordanis Kerenidis and Anupam Prakash published a quantum algorithm to achieve this task in time $\mathcal{O}(\text{poly}\log(mn))$. That is, an exponential speed-up.
- Then in 2018 Ewin Tang published a classical algorithm inspired by Kerenidis and Prakash's quantum algorithm that also achieves the task in time $\mathcal{O}(\text{poly}\log(mn))$.

Therefore, even in a classical world, there is merit in thinking quantumly.