Quantum Algorithms

Particle and Wave Aspects

Apoorva D. Patel

IISc Quantum Technology Initiative, Bangalore
http://iqti.iisc.ac.in/
Visiting Professor, ICTS-TIFR, Bangalore

Webinar Series on Quantum Algorithms using Qniverse
C-DAC Bengaluru (online)
9 July 2025

Abstract

The fact that quantum algorithms can solve some computational problems more efficiently than their classical counterparts has been the driving force behind the intense effort to develop quantum technology. I will give an overview of various quantum algorithms invented over the years, highlighting the mathematical features that provide the quantum advantage and the physical properties underpinning them. In particular, I will point out how particle and wave features contribute to them.

Quantum Technology

"Because the nature isn't classical, damn it . . ." —R.P. Feynman

Quantum Technology

"Because the nature isn't classical, damn it ..." —R.P. Feynman

It is inevitable:

The race for large scale integration and miniaturisation of computer circuits (parametrised by Moore's law) has already reached the nanoscale and the atomic scale is not very far.

Quantum effects (e.g. tunnelling) have been included in the hardware design. Incorporation of quantum logic operations is an obvious extension.

Quantum Technology

"Because the nature isn't classical, damn it ..." —R.P. Feynman

It is inevitable:

The race for large scale integration and miniaturisation of computer circuits (parametrised by Moore's law) has already reached the nanoscale and the atomic scale is not very far.

Quantum effects (e.g. tunnelling) have been included in the hardware design. Incorporation of quantum logic operations is an obvious extension.

It is a breakthrough:

Quantum logic (based on linear algebra with complex numbers) is more powerful than Boolean logic (based on integers).

Simultaneous exploitation of particle and wave properties of quantum components can substantially improve the efficiency of algorithms.

Quantum computers are hard to simulate with classical devices.

Quantum Technology

"Because the nature isn't classical, damn it . . ." —R.P. Feynman

It is inevitable:

The race for large scale integration and miniaturisation of computer circuits (parametrised by Moore's law) has already reached the nanoscale and the atomic scale is not very far.

Quantum effects (e.g. tunnelling) have been included in the hardware design. Incorporation of quantum logic operations is an obvious extension.

It is a breakthrough:

Quantum logic (based on linear algebra with complex numbers) is more powerful than Boolean logic (based on integers).

Simultaneous exploitation of particle and wave properties of quantum components can substantially improve the efficiency of algorithms.

Quantum computers are hard to simulate with classical devices.

The aim is to find and study problems that are in the computational complexity class BQP (Bounded error Quantum Polynomial).

Factorisation

Discreteness is a characteristic property of particles.

Physical properties can be localised in space-time.

Factorisation of a computational task using a suitable tensor product structure can reduce the resource requirement exponentially.

Factorisation

Discreteness is a characteristic property of particles.

Physical properties can be localised in space-time.

Factorisation of a computational task using a suitable tensor product structure can reduce the resource requirement exponentially.

Examples:

- A digital language with the place value system.
- Binary tree search for looking up a word in a dictionary.
- A multi-variable Boolean polynomial, $\prod_{i=1}^{n} x_i + \ldots + \sum_{i=1}^{n} a_i^{(1)} x_i + a^{(0)}$, has $N = 2^n$ terms.

In the factorised form, $\prod_{i=1}^{n}(c_i + x_i)$, it can be evaluated with $O(n)$ effort.

Factorisation can reduce the temporal resources by a factor $N/\log_2 N$, which is the maximal gain achievable in "particle-like" implementations.

Superposition

Superposition is a characteristic property of waves.

Multiple signals can coexist at a single space-time point.

In a Single-Instruction-Multiple-Data task, superposition of multiple parallel threads can reduce the resource requirement exponentially.

Superposition

Superposition is a characteristic property of waves.

Multiple signals can coexist at a single space-time point.

In a Single-Instruction-Multiple-Data task, superposition of multiple parallel threads can reduce the resource requirement exponentially.

Examples:

- Electromagnetic wave broadcasts for communications.
- A uniform superposition of $N = 2^n$ components can be created with n qubits and n rotations: $|0\rangle^{\otimes n} \longrightarrow \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n} = 2^{-n/2} \sum_{i=0}^{2^n-1} |i\rangle$

Superposition can reduce the spatial resources by a factor $N/\log_2 N$, which is the maximal gain achievable in “wave-like” implementations.

Superposition

Superposition is a characteristic property of waves.

Multiple signals can coexist at a single space-time point.

In a Single-Instruction-Multiple-Data task, superposition of multiple parallel threads can reduce the resource requirement exponentially.

Examples:

- Electromagnetic wave broadcasts for communications.
- A uniform superposition of $N = 2^n$ components can be created with n qubits and n rotations: $|0\rangle^{\otimes n} \longrightarrow \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n} = 2^{-n/2} \sum_{i=0}^{2^n-1} |i\rangle$

Superposition can reduce the spatial resources by a factor $N/\log_2 N$, which is the maximal gain achievable in "wave-like" implementations.

The caveat is that the final measurement destroys the superposition, extracting only $O(n)$ selected properties of the $O(2^n)$ output components (by interference, amplification, or otherwise).

Factorisation and Superposition

Classical algorithms can use either factorisation or superposition.
Quantum algorithms can use both simultaneously during execution,
but final measurements convert quantum amplitudes to classical results.

Factorisation and Superposition

Classical algorithms can use either factorisation or superposition.
Quantum algorithms can use both simultaneously during execution, but final measurements convert quantum amplitudes to classical results.

So the best of both strategies is possible, only when:

(a) The gains of factorisation and superposition do not overlap.

Otherwise there have to be trade-offs between the two.

(b) The output is concentrated in a few wave modes (δ-function).

Otherwise the success probability is suppressed.

Factorisation and Superposition

Classical algorithms can use either factorisation or superposition.
Quantum algorithms can use both simultaneously during execution,
but final measurements convert quantum amplitudes to classical results.

So the best of both strategies is possible, only when:
(a) The gains of factorisation and superposition do not overlap.

Otherwise there have to be trade-offs between the two.

(b) The output is concentrated in a few wave modes (δ-function).

Otherwise the success probability is suppressed.

The extent of quantum advantage achievable is problem dependent.

Shor's Algorithm (QFT)

Fourier Transform multiplies an N-component vector by an $N \times N$ matrix, which naively requires $O(N^2)$ operations. It is a unitary change of basis.

$$\sum_x f(x)|x\rangle = \sum_y F(y)|y\rangle = \sum_y \left(\frac{1}{\sqrt{N}} \sum_x e^{2\pi i xy/N} f(x) \right) |y\rangle$$

Shor's Algorithm (QFT)

Fourier Transform multiplies an N-component vector by an $N \times N$ matrix, which naively requires $O(N^2)$ operations. It is a unitary change of basis.

$$\sum_x f(x)|x\rangle = \sum_y F(y)|y\rangle = \sum_y \left(\frac{1}{\sqrt{N}} \sum_x e^{2\pi i x y/N} f(x)\right) |y\rangle$$

Let $N = 2^n$, and apply the same tricks as in FFT.

In binary notation, $x = x_{n-1} \cdot 2^{n-1} + \ldots + x_1 \cdot 2 + x_0$.

$\mathrm{frac}(\frac{xy}{N}) = y_{n-1}(.x_0) + y_{n-2}(.x_1 x_0) + \ldots + y_0(.x_{n-1} \ldots x_0)$.

Unitary rotation of QFT is: $|x\rangle \to \frac{1}{\sqrt{N}} \sum_y e^{2\pi i x y/N} |y\rangle$

$$= \frac{\left(|0\rangle + e^{2\pi i(.x_0)}|1\rangle\right)}{\sqrt{2}} \frac{\left(|0\rangle + e^{2\pi i(.x_1 x_0)}|1\rangle\right)}{\sqrt{2}} \cdots \frac{\left(|0\rangle + e^{2\pi i(.x_{n-1} \cdots x_0)}|1\rangle\right)}{\sqrt{2}}$$

Shor's Algorithm (QFT)

Fourier Transform multiplies an N-component vector by an $N \times N$ matrix, which naively requires $O(N^2)$ operations. It is a unitary change of basis.

$$\sum_x f(x)|x\rangle = \sum_y F(y)|y\rangle = \sum_y \left(\frac{1}{\sqrt{N}} \sum_x e^{2\pi i xy/N} f(x) \right) |y\rangle$$

Let $N = 2^n$, and apply the same tricks as in FFT.

In binary notation, $x = x_{n-1} \cdot 2^{n-1} + \ldots + x_1 \cdot 2 + x_0$.

$$\mathrm{frac}\left(\frac{xy}{N}\right) = y_{n-1}(.x_0) + y_{n-2}(.x_1 x_0) + \ldots + y_0(.x_{n-1} \ldots x_0).$$

Unitary rotation of QFT is: $|x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_y e^{2\pi i xy/N} |y\rangle$

$$= \frac{\left(|0\rangle + e^{2\pi i (.x_0)}|1\rangle\right)}{\sqrt{2}} \frac{\left(|0\rangle + e^{2\pi i (.x_1 x_0)}|1\rangle\right)}{\sqrt{2}} \cdots \frac{\left(|0\rangle + e^{2\pi i (.x_{n-1} \cdots x_0)}|1\rangle\right)}{\sqrt{2}}$$

Factorisation reduces QFT to n single qubit rotations.

Full factorisation gives the maximal $O(N/\log_2 N)$ gain.

Shor's Algorithm (Period Finding)

The problem of factoring a number N can be reduced to finding the period r of the function $f(x) = a^x \mod N$, with a coprime to N.

Whenever r is even, $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \mod N$.
So $(a^{r/2} - 1)$ and/or $(a^{r/2} + 1)$ has a factor in common with N.

Example: $N = 15$ and $a = 2$. $2^x \mod 15 = 1, 2, 4, 8, 16 \to 1, \ldots \ \Rightarrow \ r = 4, r/2 = 2$.
Both $(2^2 - 1) = 3$ and $(2^2 + 1) = 5$ are factors of 15. (GCD is easy to calculate.)

Shor's Algorithm (Period Finding)

The problem of factoring a number N can be reduced to finding the period r of the function $f(x) = a^x \mod N$, with a coprime to N.

Whenever r is even, $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \mod N$.
So $(a^{r/2} - 1)$ and/or $(a^{r/2} + 1)$ has a factor in common with N.

Example: $N = 15$ and $a = 2$. $2^x \mod 15 = 1, 2, 4, 8, 16 \to 1, \ldots \;\Rightarrow\; r = 4, r/2 = 2$.
Both $(2^2 - 1) = 3$ and $(2^2 + 1) = 5$ are factors of 15. (GCD is easy to calculate.)

Fourier transform for different values of x can be evaluated in parallel.
In the "period finding" problem, different values of x are processed in superposition, and the output components are cleverly combined into a single result. (Fourier transform of a constant is a δ-function.)

Period finding is possible with superposition of all the x values and a single run of QFT. So superposition also gives the maximal $O(N/\log_2 N)$ gain.

Shor's Algorithm (Period Finding)

The problem of factoring a number N can be reduced to finding the period r of the function $f(x) = a^x \bmod N$, with a coprime to N.

Whenever r is even, $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$.
So $(a^{r/2} - 1)$ and/or $(a^{r/2} + 1)$ has a factor in common with N.

Example: $N = 15$ and $a = 2$. $2^x \bmod 15 = 1, 2, 4, 8, 16 \to 1, \ldots \;\Rightarrow\; r = 4, r/2 = 2$.
Both $(2^2 - 1) = 3$ and $(2^2 + 1) = 5$ are factors of 15. (GCD is easy to calculate.)

Fourier transform for different values of x can be evaluated in parallel.
In the "period finding" problem, different values of x are processed in superposition, and the output components are cleverly combined into a single result. (Fourier transform of a constant is a δ-function.)

Period finding is possible with superposition of all the x values and a single run of QFT. So superposition also gives the maximal $O(N/\log_2 N)$ gain.

Factorisation over y and superposition over x are completely independent. With both gains attaining their maximal values, the algorithmic complexity reduces from $O(N^2)$ to $O((\log_2 N)^2)$.

Grover's Algorithm (Factorised)

"Database search" is a relativised problem to search for a specific item in a database using binary oracle queries.

In absence of any structure, random pickings give $\langle Q \rangle = N$.

Grover's Algorithm (Factorised)

"Database search" is a relativised problem to search for a specific item in a database using binary oracle queries.

In absence of any structure, random pickings give $\langle Q \rangle = N$.

The digital strategy for improving the process is to factorise the oracle query into smaller parts, and then sort the database in the order of the query parts. (Effort of sorting is not counted in search complexity.)
A binary search tree achieves maximal factorisation with $Q = \log_2 N$.

Grover's Algorithm (Factorised)

"Database search" is a relativised problem to search for a specific item in a database using binary oracle queries.

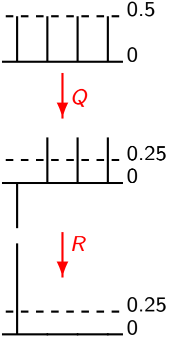
In absence of any structure, random pickings give $\langle Q \rangle = N$.

The digital strategy for improving the process is to factorise the oracle query into smaller parts, and then sort the database in the order of the query parts. (Effort of sorting is not counted in search complexity.)
A binary search tree achieves maximal factorisation with $Q = \log_2 N$.

Subsequent parallelism can only be over possibilities addressed by each query factor. Wave dynamics can uniquely identify four objects using a single binary query.
The additional gain of superposition is then $\log_2 4 = 2$.

Grover's Algorithm (Single Query)

Using a single oracle call, the algorithm identifies 1 out of 4 items in the database. In contrast, a Boolean algorithm identifies only 1 out of 2 items.
The key components are reflection operations and wave dynamics.



	Amplitudes	Algorithmic Steps	Physical Implementation
(1)		Uniform distribution	Equilibrium configuration
	Q	Quantum oracle	Binary question
(2)		Amplitude of desired state flipped in sign	Sudden perturbation
	R	Reflection about average	Overrelaxation
(3)		Desired state reached	Opposite end of oscillation
(4)	Observation	Algorithm is stopped	Measurement

(The first item is picked by the oracle. Dashed line denotes the average amplitude.)

Grover's Algorithm (Unsorted)

For an unsorted database, the oracle query cannot be factorised, and the only gain available is from superposition.

Amplitude amplification relies on clever interference, and does not have the SIMD structure. Then the achievable gain of superposition is $O(\sqrt{N})$ (not the maximal value $N/\log_2 N$).

Grover's algorithm can be executed by classical coherent wave modes, with time complexity $Q = O(\sqrt{N})$, and space complexity N.

In the mechanical setting, $|Amplitude|^2$ represents energy instead of probability.

Grover's Algorithm (Unsorted)

For an unsorted database, the oracle query cannot be factorised, and the only gain available is from superposition.

Amplitude amplification relies on clever interference, and does not have the SIMD structure. Then the achievable gain of superposition is $O(\sqrt{N})$ (not the maximal value $N/\log_2 N$).

Grover's algorithm can be executed by classical coherent wave modes, with time complexity $Q = O(\sqrt{N})$, and space complexity N.

In the mechanical setting, $|Amplitude|^2$ represents energy instead of probability.

The problem does not have two independent factors of N in its structure for factorisation and superposition to act on independently. The overlap between the two strategies limits the maximal gain to:

$$2N/\log_2 N = \begin{cases} (N/\log_2 N) \times 2 & : \text{first F then S} \\ (N/\sqrt{N}) \times (\sqrt{N}/\log_2 \sqrt{N}) & : \text{first S then F} \end{cases}$$

Grover's Algorithm (Resources)

The physical requirements for the execution of Grover's algorithm are:

(1) An initial state that is correlated in phase among its wave modes.

A tiny coupling can drive coupled oscillators to a synchronised equilibrium state.

(2) A reflection oracle that singles out the target state.

When an impurity is a node for wave propagation, the reflected wave amplitude changes sign.

(3) Coherent oscillations of the wave modes about the direction specified by the initial state.

Perturbations naturally produce oscillations about the equilibrium state.

(4) A trigger that stops the algorithm when the target state amplitude becomes sufficiently large.

There exist many phenomena and reactions that complete when a critical threshold is crossed.

All these features are also fairly immune to variations.

Grover's Algorithm (Resources)

The physical requirements for the execution of Grover's algorithm are:

(1) An initial state that is correlated in phase among its wave modes.

A tiny coupling can drive coupled oscillators to a synchronised equilibrium state.

(2) A reflection oracle that singles out the target state.

When an impurity is a node for wave propagation, the reflected wave amplitude changes sign.

(3) Coherent oscillations of the wave modes about the direction specified by the initial state.

Perturbations naturally produce oscillations about the equilibrium state.

(4) A trigger that stops the algorithm when the target state amplitude becomes sufficiently large.

There exist many phenomena and reactions that complete when a critical threshold is crossed.

All these features are also fairly immune to variations.

The factorisation advantage ($\log_2 N$ vs. N) and the superposition advantage ($O(\sqrt{N})$ vs. N) can be comparable for small N.

Note that $\sqrt{N} \leq \log_2 N$ for $N \in [4, 16]$!

For $N = 4$: Quantum algorithm needs 2 qubits and 1 oracle call.
Boolean algorithm needs 2 bits and 2 oracle calls.
Classical wave algorithm needs 4 modes and 1 oracle call.

When quantum dynamics is fragile, what is more affordable, time or space?

Biological systems exploit large scale parallelisation to gain in time.

Quantum Walks

Random walks (diffusive processes) are used to explore large spaces.
The classical diffusion operator is the Laplacian: $\frac{\partial f}{\partial t} = \nabla^2 f$

A mode with wave vector $\vec{k}$ evolves as $\exp(-E(\vec{k})t)$, with $E(\vec{k}) \propto |\vec{k}|^2$.
This non-relativistic particle-like dispersion produces the characteristic Brownian motion signature: $distance \propto \sqrt{time}$

Quantum Walks

Random walks (diffusive processes) are used to explore large spaces.
The classical diffusion operator is the Laplacian: $\frac{\partial f}{\partial t} = \nabla^2 f$

A mode with wave vector $\vec{k}$ evolves as $\exp(-E(\vec{k})t)$, with $E(\vec{k}) \propto |\vec{k}|^2$.
This non-relativistic particle-like dispersion produces the characteristic Brownian motion signature: *distance* $\propto \sqrt{time}$

Waves spread out in space according to: $\frac{\partial^2 f}{\partial t^2} = \nabla^2 f$

A mode with wave vector $\vec{k}$ evolves as $\exp(iE(\vec{k})t)$, with $E(\vec{k}) \propto |\vec{k}|$.
This relativistic evolution produces the signature: *distance* $\propto$ *time*

Quantum Walks

Random walks (diffusive processes) are used to explore large spaces.
The classical diffusion operator is the Laplacian: $\frac{\partial f}{\partial t} = \nabla^2 f$

A mode with wave vector $\vec{k}$ evolves as $\exp(-E(\vec{k})t)$, with $E(\vec{k}) \propto |\vec{k}|^2$.
This non-relativistic particle-like dispersion produces the characteristic Brownian motion signature: $distance \propto \sqrt{time}$

Waves spread out in space according to: $\frac{\partial^2 f}{\partial t^2} = \nabla^2 f$

A mode with wave vector $\vec{k}$ evolves as $\exp(iE(\vec{k})t)$, with $E(\vec{k}) \propto |\vec{k}|$.
This relativistic evolution produces the signature: $distance \propto time$

Quantum theory can be successfully combined with special relativity.
The coin appears as the particle-antiparticle internal degree of freedom.
The flip-flop walk contains a Klein-Gordon propagation mode.

Any NP-complete problem can be speeded up at least quadratically.

References

A. Patel, *Quantum Computation: Particle and Wave Aspects of Algorithms*, Resonance 16 (2011) 821-835, `arXiv:1108.1659`.

A. Patel, *Grover's Algorithm in Natural Settings*, Quantum Inf. Comput. 21 (2021) 945-954, `arXiv:2001.00214`.

A. Patel and M.A. Rahaman, *Search on a Hypercubic Lattice using Quantum Random Walk:* $d > 2$, Phys. Rev. A 82 (2010) 032330, `arXiv:1003.0065`.

A. Patel and A. Priyadarsini, *Optimisation of Quantum Hamiltonian Evolution: From Two Projection Operators to Local Hamiltonians*, Int. J. Quantum Inf. 15 (2017) 1650027, `arXiv:1503.01755`.

T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H.S. Derks, P.K. Faehrmann and J.J. Meyer, *Training Quantum Embedding Kernels on Near-Term Quantum Computers*, `arXiv:2105.02276` (2021).

H.-Y. Huang, R. Kueng and J. Preskill, *Information-Theoretic Bounds on Quantum Advantage in Machine Learning*, Phys. Rev. Lett. 126 (2021) 190505, `arXiv:2101.02464` (2021).