Grover's Algorithm & Quantum Monte Carlo Integration

Steven Herbert

1.0 INTRODUCTION

In 1996 Indian-American computer scientist *Lov Grover* published a quantum search algorithm, which to this day remains one of the most important quantum algorithms. The algorithm concerns searching an *unstructured database* with N entries. If we are searching for a unique *marked* entry, then classically this would take a maximum of N queries, and $N/2$ queries on average. Grover's algorithm enables the task to be completed with only $\mathcal{O}(\sqrt{N})$ queries. This is commonly referred to as a 'quadratic advantage', and Grover's algorithm underpins many of algorithms that are anticipated to lead to early quantum advantage in real-world applications.

2.0 QUANTUM COMPUTING 101

2.1 Qubits

A classical bit is an intuitive concept, it is either equal to:

$$0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \text{ or } 1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

Even if we are uncertain about whether a classical bit, B, is in state 0 or 1, we can characterise it by a probability mass function, or a *mixture*

$$p(B = 0) = p_0 \ ; \ p(B = 1) = p_1$$

where $p_0 + p_1 = 1$. A qubit, $|\psi\rangle$, is quite different, it can be in a *superposition* of the 0 and 1 states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$.

The first thing to note about a qubit is that we can still extract *classical information* by *measuring* the qubit. For example measuring the state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

results in $|0\rangle$ with probability $|\alpha|^2$, and $|1\rangle$ with probability $|\beta|^2$. This is known as the *Born rule* (after Max Born). After the measurement, the system is in the measured state. That is, the post-measurement state, $|\psi'\rangle$, will be:

$$|\psi'\rangle = |0\rangle \text{ or } |\psi'\rangle = |1\rangle$$

Further measurements will always yield the same value (assuming no quantum operations are applied to the qubit between the measurements). This means that we can only extract one bit of information from the state of a qubit.

The superposition of $|0\rangle$ and $|1\rangle$ states describes a physical structure, and not merely a probability mass function over possible measurement outcomes. For example:

- $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$
- $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$
- $\frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) = |i\rangle$
- $\frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle) = |-i\rangle$

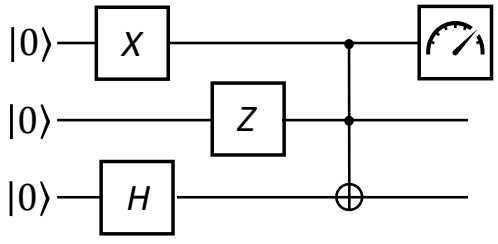
all have a 50% chance of being in either in the $|0\rangle$ or $|1\rangle$ state if measured, but all correspond to different superpositions, which will evolve differently when quantum operations are applied. It is crucial to appreciate this point to grasp the essence of quantum computing.

2.2 Quantum circuits

A quantum circuit is a tensor network of n qubits, with three stages:

- Initialisation of all qubits in the $|0\rangle$ state (denoted $|0\rangle^{\otimes n}$).
  - Sometimes it will be convenient to let the initial state be something other than $|0\rangle^{\otimes n}$ – but we should be able to efficiently prepare this initial state from $|0\rangle^{\otimes n}$ (that is, using a number of one- and two-qubit operations that is at most polynomial in the number of qubits).
- Some quantum gates, which represent unitary transformations.
- A final layer of measurements in the computational basis, on some or all of the qubits.
  - In fact, by the *principle of implicit measurement*, we can consider **all** qubits to be measured in the final layer.

An example quantum circuit:



2.3 Quantum operations

For the algorithms we study here, it is most important to be familiar with:

- The *Hadamard* gate, which prepares an equal superposition when applied to a computational basis state:

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |-\rangle$$

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

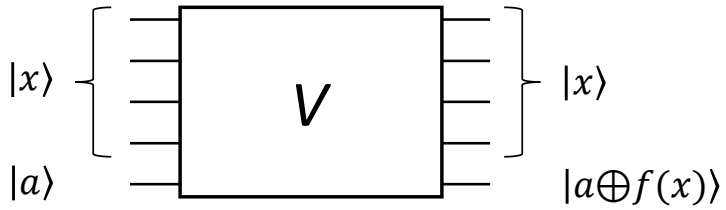
- The *Pauli-X* gate, which is a not gate:

$$X|0\rangle = |1\rangle \qquad X|1\rangle = |0\rangle$$

  and in general may be *controlled* by the state of one or more other qubits.

3.0 ORACLE CIRCUITS

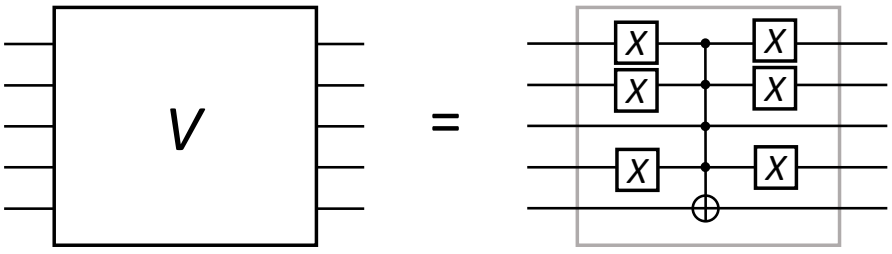
An "oracle" can be thought of as something that can *recognise* a correct answer when it sees one. Specifically, we think of an oracle as a black box "marking" some input binary strings as 0 (i.e., $f(x) = 0$) and some as 1 (i.e., $f(x) = 1$), of the form:



- In Grover's algorithm, we consider a *search oracle*, V, which marks a single element as 1, and all the others as 0.
- The goal is to find the element marked 1, and we assume that there is no algorithmic short-cut to find it, so classically we would have to perform a brute-force search.

3.1 An example of an oracle

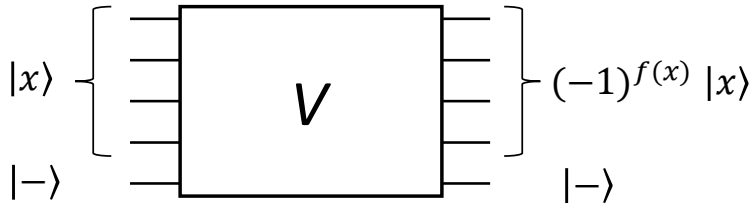
Say that 0010 is the unique "marked" binary string, then an appropriate oracle would be:



Which uses a multi-controlled not gate. Specifically, the gate sandwiched between the two layers of X gates denotes a multi-controlled X (not) gate, where a Pauli-X is applied to the final qubit when the first four qubits are equal to 1.

3.2 The action of a search oracle on $|-\rangle$

It is particularly important to understand what happens when the final qubit is initialised in the state $|-\rangle$, which achieves an effect sometimes referred to as 'phase-kickback'.



- Consider an input binary string x and let the final qubit input be set as $|-\rangle$, (i.e., the input is $|x\rangle\,|-\rangle$)
- The oracle transforms this to $(-1)^{f(x)}\,|x\rangle\,|-\rangle$. That is, if $f(x) = 0$ then nothing happens, whereas if $f(x) = 1$ then the last qubit is changed from $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ to $\frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ because of the modulo-2 addition.

3.3 Decomposing the uniform superposition

Grover's algorithm requires that the search oracle is applied not to a single input state, $|x\rangle$, but rather to a uniform superposition of all input bitstrings of length $n = \log_2 N$ (for convenience we let the number of elements in the search space, N, be a power of 2). This is achieved by initially applying $H^{\otimes n}$ to the search register to *prepare* the state:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle$$

Which we can write:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \left(\sqrt{N-1}\, \underbrace{\frac{1}{\sqrt{N-1}} \sum_{x \text{ s.t. } f(x)=0} |x\rangle}_{|y\rangle} + |z\rangle \right)$$

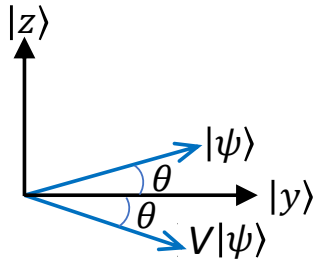
where $|z\rangle$ is the single basis state $|x\rangle$ such that $f(x) = 1$.

3.4 Visualising the action of a search oracle on $|-\rangle$

Now we consider the effect of the search oracle on this input, as can be visualised in a two-dimensional space where (as defined above):

- $|y\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \text{ s.t.} f(x)=0} |x\rangle$, i.e., a unit vector in the direction of the uniform superposition of all unmarked strings.

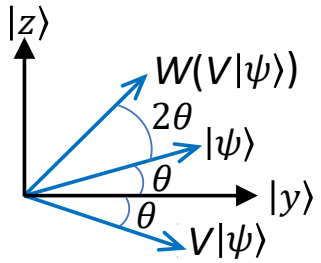
- $|z\rangle = |x\rangle$ where $f(x) = 1$, i.e., the marked element.



Applying the search oracle to $|\psi\rangle$ does nothing to the component in the $|y\rangle$ direction, but multiplies the component in the $|z\rangle$ direction by -1. So the action of the search oracle can be seen as a reflection in the $|y\rangle$ axis.

3.5 The second step: reflecting about the original superposition

As we have seen above, the search oracle leads to a reflection in the $|y\rangle$ axis. However to find the marked element we need to rotate the superposition towards the $|z\rangle$ axis to increase the probability of measuring the marked state (i.e., x s.t. $f(x) = 1$). So we follow up the oracle step with another reflection, about the line of the original uniform superposition, which is given by $W = (2\,|\psi\rangle\langle\psi| - I)$ where $|\psi\rangle = |+\rangle^{\otimes n}$. This second reflection can also be depicted:



To see that the unitary W does indeed perform the reflection as claimed, consider an arbitrary (real) vector $|\phi\rangle$, decomposed into a component in the direction of the uniform superposition (denoted $|\psi\rangle$ as previously defined), and a component perpendicular to the uniform superposition, which we denote $|\psi^\perp\rangle$:

$$|\phi\rangle = a\,|\psi\rangle + b\,|\psi^\perp\rangle$$

So it follows:

$$\begin{aligned}
W\,|\phi\rangle =& (2\,|\psi\rangle\langle\psi| - I)\,|\phi\rangle \\
=& (2\,|\psi\rangle\langle\psi| - I)(a\,|\psi\rangle + b\,|\psi^\perp\rangle) \\
=& 2a\,|\psi\rangle \underbrace{\langle\psi|\psi\rangle}_{=1} - 2b\,|\psi\rangle \underbrace{\langle\psi|\psi^\perp\rangle}_{=0} - a\,|\psi\rangle - b\,|\psi^\perp\rangle \\
=& 2a\,|\psi\rangle - a\,|\psi\rangle - b\,|\psi^\perp\rangle \\
=& a\,|\psi\rangle - b\,|\psi^\perp\rangle
\end{aligned}$$

That is, the desired reflection about the uniform superposition.

To complete the argument, it is necessary to further show that W can be implemented with gates from a standard quantum gate-set. We start by decomposing $W = 2\left|\psi\right\rangle\left\langle\psi\right| - I$ (where $\left|\psi\right\rangle = \left|+\right\rangle^{\otimes n}$):

$$\begin{aligned} H^{\otimes n} W H^{\otimes n} =& 2\left|0^n\right\rangle\left\langle 0^n\right| - H^{\otimes n} I H^{\otimes n} \\ =& 2\left|0^n\right\rangle\left\langle 0^n\right| - I \end{aligned}$$

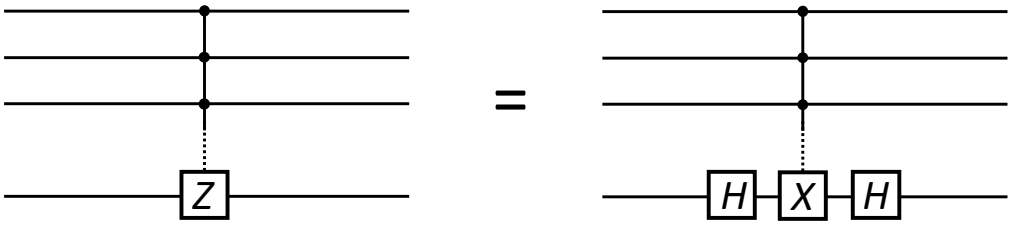
because H is self-inverse, and $H\left|+\right\rangle = \left|0\right\rangle$ (note $\left|0^n\right\rangle = \left|0\right\rangle^{\otimes n}$). We can also use the fact that $X\left|0\right\rangle = \left|1\right\rangle$:

$$\begin{aligned} X^{\otimes n}(H^{\otimes n} W H^{\otimes n})X^{\otimes n} =& X^{\otimes n}(2\left|0^n\right\rangle\left\langle 0^n\right| - I)X^{\otimes n} \\ =& 2\left|1^n\right\rangle\left\langle 1^n\right| - I \end{aligned}$$

which is the matrix:

$$2\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix} = -\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & -1 \end{bmatrix}$$

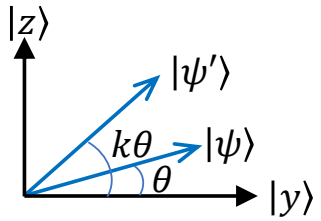
We have that $(-2\left|1^n\right\rangle\left\langle 1^n\right| + I)$ is a n-qubit generalisation of the CZ gate, which we denote $C_{n-1}Z$:



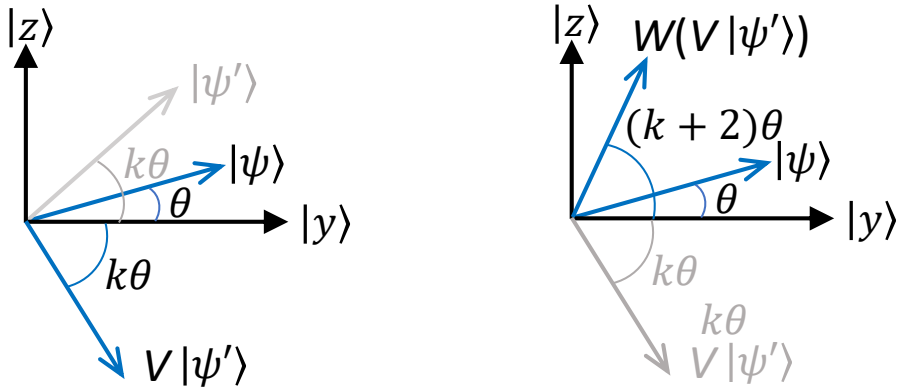
As $Z = HXH$, we have that $C_{n-1}Z = (I \otimes H)C_{n-1}X(I \otimes H)$, which can be efficiently implemented using Toffoli gates with some extra workspace qubits. Putting this altogether we have:

$$W = -H^{\otimes n} X^{\otimes n}(I \otimes H)C_{n-1}X(I \otimes H)X^{\otimes n} H^{\otimes n}$$

We have shown that following the search oracle, V, with the unitary W achieves a rotation of 2θ. It is crucial to appreciate that this rotation is achieved if the state is initially at *any* starting angle, $k\theta$. That is:



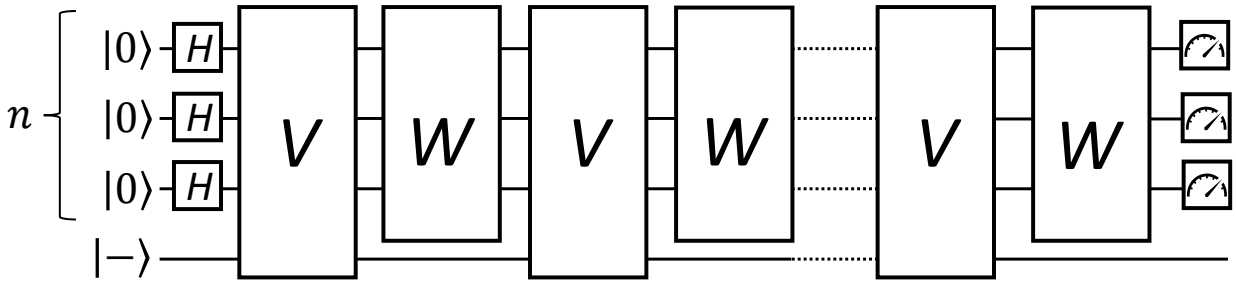
Then the action of the matrices V and then W will be:



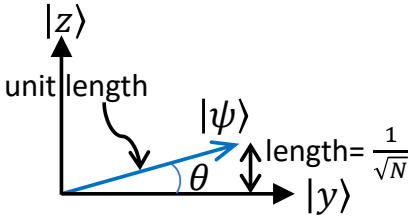
$(W \otimes I)V$ is called the *Grover* iterate and leads to a 2θ rotation.

4.0 GROVER'S ALGORITHM

Grover's algorithm consists of repeated applications of the Grover iterate (V followed by W), which rotates the uniform superposition $|\psi\rangle$ towards a superposition consisting of only (or dominated by) the marked element, $|z\rangle$:



It is therefore necessary to count how many Grover iterates are required. By simple trigonometry we can express θ:

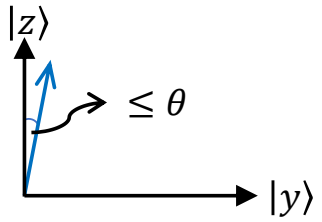


So $\sin\theta = \left(\frac{1}{\sqrt{N}}\right)/1$. If we are searching a large database then $\theta \approx \sin\theta$. Each iteration rotates the superposition 2θ, and we need to rotate $(\frac{\pi}{2} - \theta)$ in total, thus we require n_{it} iterations, where:

$$\begin{aligned} \frac{\pi}{2} - \theta =& n_{it} \times 2\theta \\ \implies n_{it} =& \frac{\pi}{4\theta} - \frac{1}{2} \\ \approx& \frac{\pi\sqrt{N}}{4} \end{aligned}$$

so we require only $\mathcal{O}(\sqrt{N})$ Grover iterates, as opposed to checking all N elements in a classical brute-force search. This *is* the quadratic advantage. Furthermore, it has been shown that $\Theta\sqrt{N}$ is a lower bound for search of an unstructured database.

It is worth noting that in general N may be such that the superposition never quite aligns with $|z\rangle$, however we can always rotate to within θ of $|z\rangle$:



Therefore we will measure the marked answer with probability at least:

$$(\cos\theta)^2 = 1 - (\sin\theta)^2 = 1 - \left(\frac{1}{\sqrt{N}}\right)^2 = \frac{N-1}{N}$$

Interpreting the quadratic advantage

- Each element in the database has a $1/N$ probability of being the marked element. It follows that if we classically search through the elements, each new element that we inspect increases the probability of us having found the marked element by $1/N$. Thus taking $1/(1/N) = N$ iterations to find the marked state with certainty.
- Quantumly, we put all elements in superposition and then "move the superposition around". Crucially, the (modulus of the) co-efficient of a term in the superposition is the square root of its probability, and so the co-efficient of the marked element is $\sqrt{1/N}$. As we move the superposition such that the co-efficient of the marked element increases in constant increments we only take $\mathcal{O}(1/(\sqrt{1/N}) = \mathcal{O}(\sqrt{N})$ iterations to find the marked state.

What happens if there is more than one marked state?

If we know that there are M solutions (M marked elements), then the same analysis can be applied to the case in which $|z\rangle$ is not a single marked element, but the equal superposition of *all* marked elements, and yields $\frac{\pi}{4}\sqrt{N/M}$ iterations needed to find *some* solution.

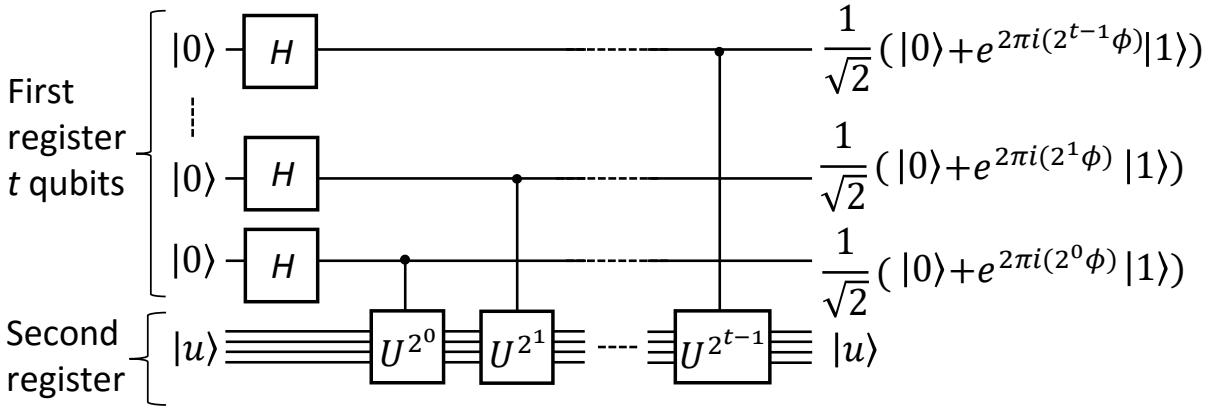
But what about if we don't even know M? This is a much trickier question, and to understand the answer we must first introduce *quantum phase estimation*.

5.0 QUANTUM PHASE ESTIMATION

Quantum phase estimation addresses the following problem:

- We have a n-qubit oracle function U, encoded in the form of a controlled-U unitary.
- U has an eigenvalue $e^{2\pi i\phi}$, associated with an eigenvector $|u\rangle$ which we can prepare.
- We wish to estimate the phase, ϕ, of the eigenvalue to t bits of precision.

This can be achieved by the following circuit:



The final state can be expressed:

$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i\phi j} |j\rangle |u\rangle$$

The inverse *quantum Fourier transform* is then applied to the first register, which achieves:

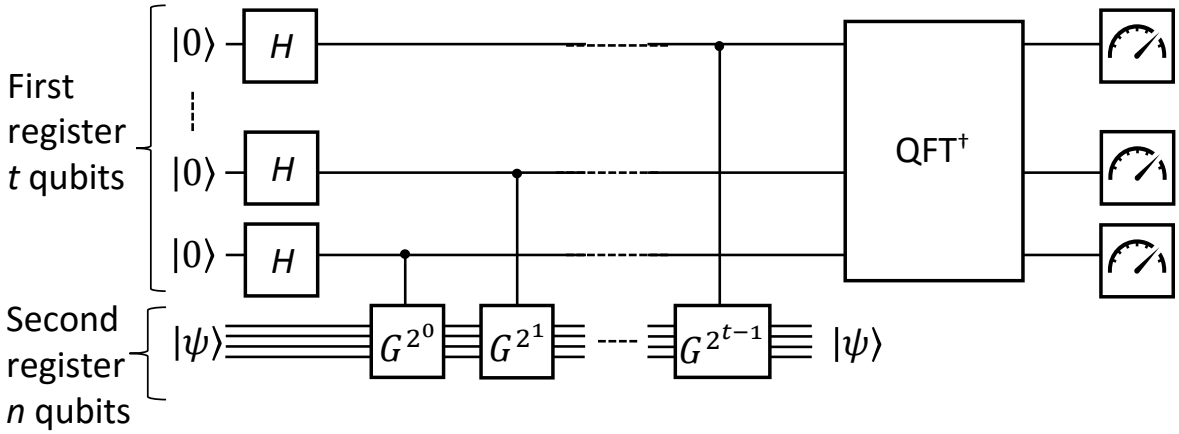
$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i\phi j} |j\rangle |u\rangle \rightarrow |\tilde{\phi}\rangle |u\rangle$$

Thus measuring the first register, whose state is now $|\tilde{\phi}\rangle$, gives a t-bit approximation, $\tilde{\phi}$, of the eigenvalue phase, ϕ.[1]

[1]The specifics of how quantum phase estimation works are outside of the scope of these notes, however further details can be found here: https://www.cl.cam.ac.uk/teaching/2122/QuantComp – which form the basis of a forthcoming textbook on Quantum Computing.

Quantum phase estimation for approximate quantum counting

If we let the Grover iterate be G, i.e., $G = (W \otimes I)V$, then we can 'approximately count' M using the following circuit.



To see how this works, recall that the effect of the Grover iterate is to rotate by an angle 2θ in the plane spanned by $|y\rangle$ and $|z\rangle$. If we consider the projection onto this plane, and define $|y\rangle = [1, 0]^T$ and $|z\rangle = [0, 1]^T$, we have that:

$$G = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

from which we can verify that G has:

- an eigenvalue $e^{2i\theta}$ with eigenvector

  $$(1/\sqrt{2})[i, 1]^T = (1/\sqrt{2})(i\,|y\rangle + |z\rangle)$$

  which we define as $|e_1\rangle$;

- and also an eigenvalue $e^{i(2\pi - 2\theta)}$ with eigenvector

  $$(1/\sqrt{2})[-i, 1]^T = (1/\sqrt{2})(-i\,|y\rangle + |z\rangle)$$

  which we define as $|e_2\rangle$.

Therefore:

$$|\psi\rangle = \frac{\sqrt{N-M}}{\sqrt{N}}\,|y\rangle + \frac{\sqrt{M}}{\sqrt{N}}\,|z\rangle = a\,|e_1\rangle + b\,|e_2\rangle$$

for some a and b.

Depending on which eigenvector the measurement collapses to, the output will be an estimate of θ/π or $(\pi - \theta)/\pi$ (the eigenvalue phases) with maximum error 2^{-t}. We now must analyse how much this round-off

error in the phase estimation affects our estimate of M. Specifically, we will express the error in the estimate of M, ϵ_M, in terms of the error in the estimate of θ, ϵ_θ, and hence t. If there are M (rather than one) marked states, we have that $\sin\theta = \sqrt{M/N}$. Thus we have:

$$\frac{\epsilon_M}{N} = \sin^2(\theta + \epsilon_\theta) - \sin^2(\theta)$$

We can re-arrange this as:

$$\frac{\epsilon_M}{N} = (\sin(\theta + \epsilon_\theta) - \sin(\theta))\,(\sin(\theta + \epsilon_\theta) + \sin(\theta))$$

which allows us to bound the maximum absolute value of ϵ_M:

$$\frac{|\epsilon_M|}{N} = |\sin(\theta + \epsilon_\theta) + \sin(\theta)| \times |\sin(\theta + \epsilon_\theta) - \sin(\theta)|$$

As the magnitude of the gradient of $\sin\theta$ is at most one, the second term in the right-hand side can be upper-bounded by $|\epsilon_\theta|$. Using a simple result from trigonometry $|\sin(\theta + \epsilon_\theta)| < \sin\theta + |\epsilon_\theta|$ (noting that the problem set-up is such that $0 \leq \theta \leq \pi/2$, so $\sin\theta$ is non-negative) and also $M/N = \sin^2\theta$ and $\epsilon_\theta \leq \pi 2^{-t}$ (the factor of π because we are performing phase estimation, so estimate θ/π) we get:

$$\begin{aligned}
\frac{|\epsilon_M|}{N} &< |2\sin\theta + \epsilon_\theta| \times |\epsilon_\theta| \\
&< \left(2\sqrt{M/N} + \frac{\pi}{2^t}\right)\frac{\pi}{2^t} \\
\implies |\epsilon_M| &< \left(2\sqrt{MN} + \frac{\pi N}{2^t}\right)\frac{\pi}{2^t}
\end{aligned}$$

finally, choosing $t = n/2$, i.e., $2^t = \sqrt{2^n} = \sqrt{N}$ we get:

$$\begin{aligned}
|\epsilon_M| &< \left(2\sqrt{MN} + \pi\sqrt{N}\right)\frac{\pi}{\sqrt{N}} \\
&= \pi\left(2\sqrt{M} + \pi\right)
\end{aligned}$$

So we have that we can estimate M with maximum error $\mathcal{O}(\sqrt{M})$, and we can see that for $t = n/2$ we must call the Grover operate G a total of $2^{n/2} = \sqrt{N}$ times. Therefore we only require $\mathcal{O}(\sqrt{N})$ Grover iterations to approximately count M. Classically we would require N operations to do this exactly, and it turns out that we would still need $\mathcal{O}(N)$ operations to approximately classically count M to the same accuracy as we have achieved quantumly.

6.0 GENERAL QUANTUM AMPLITUDE ESTIMATION

We have interpreted the phase estimation of the Grover iterate as *counting* the number of marked elements. However, we could equally have thought of it as *estimating* its amplitude, $\sin^2\theta$. When considering amplitude estimation, it is generally convenient to think of the error in terms of the *mean-squared error* (MSE) which is $\mathcal{O}(q^{-2})$, where q is the number of Grover iterates.

Quantum amplitude estimation applies to more than just Grover iterates built from a search oracle (as studied above), to see this note that *every* quantum state can be written in the form:

$$|\psi\rangle = \cos\theta\,|\Psi_0\rangle\,|0\rangle + \sin\theta\,|\Psi_1\rangle\,|1\rangle$$

for some angle θ and quantum states $|\Psi_0\rangle$ and $|\Psi_1\rangle$. We let A be the circuit that *prepares* $|\psi\rangle$, that is:

$$A\,|\mathbf{0}\rangle = |\psi\rangle$$

and with two uses of A a general 'Grover iterate', Q can be constructed such that:

$$Q^m\,|\psi\rangle = \cos((2m+1)\theta)\,|\Psi_0\rangle\,|0\rangle + \sin((2m+1)\theta)\,|\Psi_1\rangle\,|1\rangle$$

Thus we can use quantum phase estimate to estimate the amplitude, $\sin^2\theta$ of a general quantum state, $|\psi\rangle$.

Amplitude estimation without phase estimation

Whilst the above suffices to assert the *asymptotic* advantage of QAE over classical approaches, in practice the Quantum Fourier Transform is an infeasibly large circuit to run on real near-term hardware. It was therefore a massive break-through when *amplitude estimation without phase estimation* was proposed [1]. There are now a few flavours for how to do this, but they all have the same essential form:

- Prepare $Q^m\,|\psi\rangle$ for some m.
- Measure several 'shots' of this circuit.
- The outcome 1 is measured with probability $\sin^2((2m+1)\theta)$
- Repeat for a variety of values of m
- Use (Bayesian) post-processing to infer an estimate of θ from the measurement outcomes.

For an exponentially increasing sequence of m ($m = \{0, 1, 2, 4, 8, 16, \dots\}$), and constant number of shots for each m, it was shown that the full-quadratic advantage is achieved.

We now move on from the quantum algorithm itself, and see how quantum amplitude estimation can be used as a subroutine to speed-up *Monte Carlo integration*.

7.0 MONTE CARLO INTEGRATION

The expectation (mean) of a random variable can be estimated by averaging samples (let $X \sim p(x)$):

$$\mathbb{E}(X) = \int xp(x)\mathrm{d}x \approx \frac{1}{N}\sum_{j=1}^{N} X_j$$

When performed on a digital computer (classical or quantum), any continuous $p(x)$ must be truncated and discretised:

$$\mathbb{E}(X) = \sum_x xp(x) \approx \frac{1}{N}\sum_{j=1}^{N} X_j$$

And in general we may want to estimate the expectation when a function is applied to the samples:

$$\mathbb{E}(f(X)) = \sum_x f(x)p(x) \approx \frac{1}{N}\sum_{j=1}^{N} f(X_j)$$

7.1 Quantum Monte Carlo Integration

We assume that we have a circuit, P, that prepares a quantum state, $|p\rangle$, encoding a probability distribution, $p(x)$ (note that all of the following also applies to multivariate distributions). That is, $|p\rangle = P|\mathbf{0}\rangle$ such that:

$$|p\rangle = \sum_x \sqrt{p(x)}\,|x\rangle$$

We also have a circuit, R, that operates on $|p\rangle|0\rangle$ such that:

$$R|p\rangle|0\rangle = \sum_x \sqrt{p(x)}\,|x\rangle\left(\sqrt{1-f(x)}\,|0\rangle + \sqrt{f(x)}\,|1\rangle\right)$$

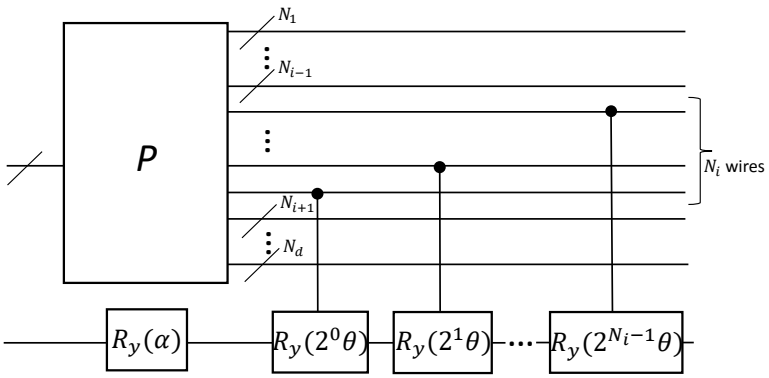
We can now see that the probability of measuring 1 on the final qubit is:

$$\sum_x p(x)f(x)$$

This can be computed using quantum amplitude estimation (QAE).

The key question to ask now is: *what is R?* In fact, R will only be a simple circuit if f is a trigonometric function.

[2, Proposition 2]. Let $A(\beta)$ be the circuit



Reproduced from [2, Fig. 1]

where $\alpha = n\omega x_l - \beta$ and $\theta = n\omega\Delta$. Then:

i. $1 - 2\texttt{QAE}(A(0))$ is an estimate of:

$$\sum_x p(x)\cos(n\omega x)$$

ii. $1 - 2\texttt{QAE}(A(\pi/2))$ is an estimate of:

$$\sum_x p(x)\sin(n\omega x)$$

That is, in this case, R is just a bank of controlled rotations.

7.2 QMCI: the full advantage in minimal circuit depth

In general the circuit R may be thought of as performing arithmetic operations on the quantum computer, and we now see how the minimamally deep circuit for R, that is R being just a bank of rotation gates, can be used to computer *arbitrary* Monte Carl integrals. This is summarised in [2, Table 1]:

Method	Computes	MSE	Arithmetic
Classical MCI	$\mathbb{E}(f(X))$	$\Theta(q^{-1})$	Classical
Quantum MCI	$\mathbb{E}(f(X))$	$\Theta(q^{-2})$	Quantum & classical
Rescaled QMCI [3, 4]	$\mathbb{E}(X)$	$\Theta(q^{-4/3})$	Classical only
Fourier QMCI [2]	$\mathbb{E}(f(X))$	$\Theta(q^{-2})$	Classical only

[2, Theorem 3] *The quantity, $\mu = \mathbb{E}(f(X))$, where $X \sim p(x)$ can be estimated with MSE $\in \Theta(q^{-\lambda})$, where q is the number of uses of a circuit preparing the quantum state $|p\rangle$ and λ is the convergence rate of some QAE subroutine which operates on circuits of the form $A(\beta)$.*

Note that this result holds for *any* QAE algorithm, and typically (for QAE subroutines that achieve the full quadratic advantage), $\lambda = 2$.

Sketch of proof of [2, Theorem 3]

First, we build a piecewise periodic function $\mathrm{f}(x)$ such that $\mathrm{f}(x) = f(x)$ over the support of $p(x)$ (defined as the interval $x_l \leq x < x_u$):

$$\mathrm{f}(x) = \begin{cases} f(x) & \text{if } x_l \leq x < x_u \\ \tilde{f}(x) & \text{if } x_u \leq x < x_{\tilde{u}} \end{cases}$$

$\mathrm{f}(x)$ is periodic, hence has a Fourier series decomposition. Moreover, we can always construct this such that the Fourier series coefficients decay as $1/n^3$, that is:

$$\mathrm{f}(x) = c + \sum_{n=1}^{\infty} \frac{1}{n^3} \left(\tilde{a}_n \cos(n\omega x) + \tilde{b}_n \sin(n\omega x) \right)$$

where $|\tilde{a}_n|$, $|\tilde{b}_n| \in \mathcal{O}(1)$.

As $\mathrm{f}(x) = f(x)$ whenever $p(x) \neq 0$, we can express:

$$\begin{aligned} \mu &= \sum_x p(x) f(x) = \sum_x p(x) \mathrm{f}(x) \\ &= \sum_x p(x) \left(\sum_{n=1}^{\infty} \frac{1}{n^3} \left(\tilde{a}_n \cos(n\omega x) + \tilde{b}_n \sin(n\omega x) \right) + c \right) \\ &= c + \sum_{n=1}^{\infty} \frac{\tilde{a}_n}{n^3} \left(\sum_x p(x) \cos(n\omega x) \right) + \sum_{n=1}^{\infty} \frac{\tilde{b}_n}{n^3} \left(\sum_x p(x) \sin(n\omega x) \right) \end{aligned}$$

To estimate μ, we can estimate each of the parenthesised sums individually. This can be achieved according to the procedure described in Proposition 2.

The $1/n^3$ convergence of the Fourier series is crucial as it allows:

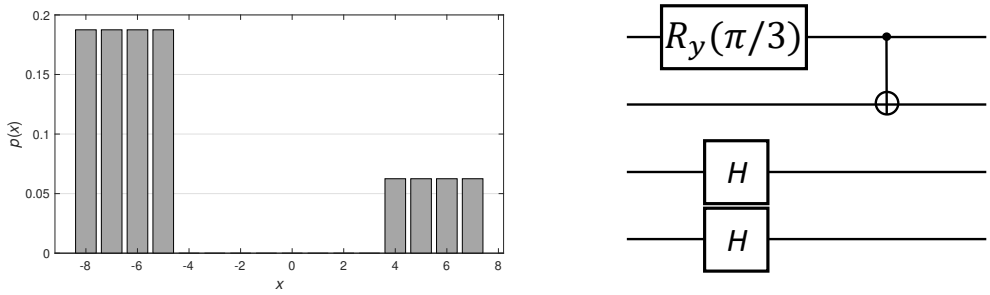
1. the uses of the circuit P to be spread amongst the various Fourier series components;

2. the sum approximating the expectation to be truncated,

such that the overall claim of convergence holds. The remainder of the proof is concerned with showing that the error is indeed as claimed.

Numerical results

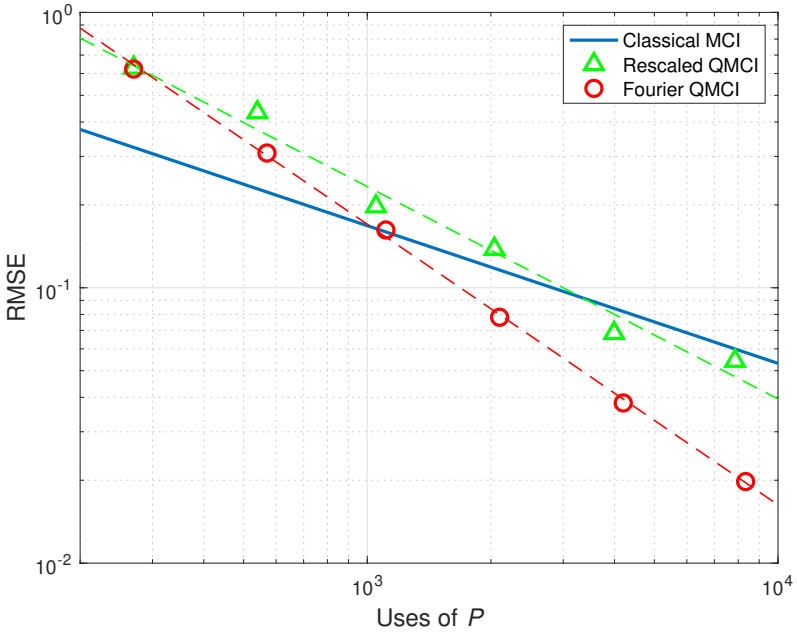
To test the effectiveness of Fourier QMCI in practice, the following test circuit was considered:



(a) (b)

[2, Fig. 2] (a) The probability distribution used for the numerical results; (b) a circuit that prepares a quantum state encoding this distribution.

which gave the following results:



[2, Fig. 3] A comparison of classical MCI; rescaled QMCI; and Fourier QMCI

8.0 APPLICATIONS

Monte Carlo Integration applications can broadly be grouped according to the following:

- **Monte Carlo Estimation**: where we wish to estimate some statistical quantity using Monte Carlo Integration.
- **Simulation-Based Optimisation**: where each cost function evaluation in an optimisation algorithm requires Monte Carlo Estimation

Fourier QMCI is particularly suited to applications posed in terms of samples from some distribution that are averaged *after* some function is applied – and using Fourier QMCI we get this function for "free" (without additional quantum operations).

Monte Carlo methods in general (and Monte Carlo *integration* in particular) are ubiquitous in myriad science, business, technology, engineering and financial application.

If...

1. A process with some randomness must be modelled (the random process may capture uncertainty about future evolution, and in this way Monte Carlo *estimation* may be used for *prediction*);
2. the process is sufficiently complex (high-dimensional); and
3. in the end, some general quantity must be extracted,

then the chances are that MCI is the best method. In particular, extraction of many quantities often amounts to *integration*, even if we are not used to thinking about it in those terms.

8.1 Application of QMCI to computational fluid dynamics

Is essence, the reason that we expect QMCI to find application to CFD boils down to the following argument:

- Partial differential equations are ubiquitous in CFD.
- When these are sufficiently high-dimensional finite-difference methods become ineffective.
- Instead Monte Carlo simulation may be used when there is randomness in the process (either intrinsically present, or injected to enable the solution to be found by a randomised approach).

This argument is essentially identical to that which has led to QMCI gaining great attention in finance: asset prices are governed by stochastic differential equations, and pay-offs are expectation values.

8.2 QMCI: opportunities for quantum advantage

- If some problem can be expressed as: sample, apply a function and then average; and the computational load is a bottleneck, then the guaranteed quadratic quantum advantage is likely to be very valuable in practice.

- The Fourier series decomposition in Fourier QMCI will enable such an advantage to be realised with minimal quantum resources.
- Additionally, as the quantum algorithm exactly puts the desired quantity onto the amplitude of a single qubit (which is thereafter estimated), QMCI performs comparably well for every statistical quantity estimated. Therefore we expect particularly early useful quantum advantage in applications such as calculating expectations pertaining to rare / extreme events.

8.3 QMCI: challenges and use-case specific questions

Whilst Fourier QMCI unquestionably provides a significant practical advantage over standard QMCI, in general, the decomposition of some random process into a 'sampled distribution' and subsequent 'applied function' is not unique. Therefore, for any specific use-case we expect to require further research to tailor the application-specific best design.

Regardless of how this decomposition is done, invariably the 'state preparation' stage will encode real data in some sense. Hence it is sometimes alternatively referred to as data-loading – which is widely accepted as one of the most prominent challenges in quantum computing in general.

8.4 QMCI: timescales to see useful quantum advantage

Google and IBM have pledged to build 1 Million qubit machines by the end of the decade (2030), which is widely anticipated to be enough qubits to see useful quantum advantage in a wide-range of areas, including many applications of QMCI. Quantinuum also has aggressive, but realistic, plans for scaling-up of hardware – and even though the promised qubit numbers will be less, the intrinsic technology (a trapped-ion rather than superconducting qubit device) is vastly superior (much longer coherence time), and so even with fewer qubits we anticipate comparable performance.

We have done specific benchmarking for applications in finance (as that is the area that has thus far received the most attention as an application of QMCI) and predict that a few thousand to a few tens of thousands of qubits will be enough to yield useful quantum advantage.

REFERENCES

[1] Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., and Yamamoto, N., "Amplitude estimation without phase estimation," *Quantum Information Processing*, Vol. 19, No. 2, Jan 2020.

[2] Herbert, S., "Quantum Monte-Carlo Integration: The Full Advantage in Minimal Circuit Depth," 2021.

[3] Woerner, S. and Egger, D. J., "Quantum risk analysis," *npj Quantum Information*, Vol. 5, No. 1, Feb 2019.

[4] Stamatopoulos, N., Egger, D. J., Sun, Y., Zoufal, C., Iten, R., Shen, N., and Woerner, S., "Option Pricing using Quantum Computers," *Quantum*, Vol. 4, Jul 2020, pp. 291.