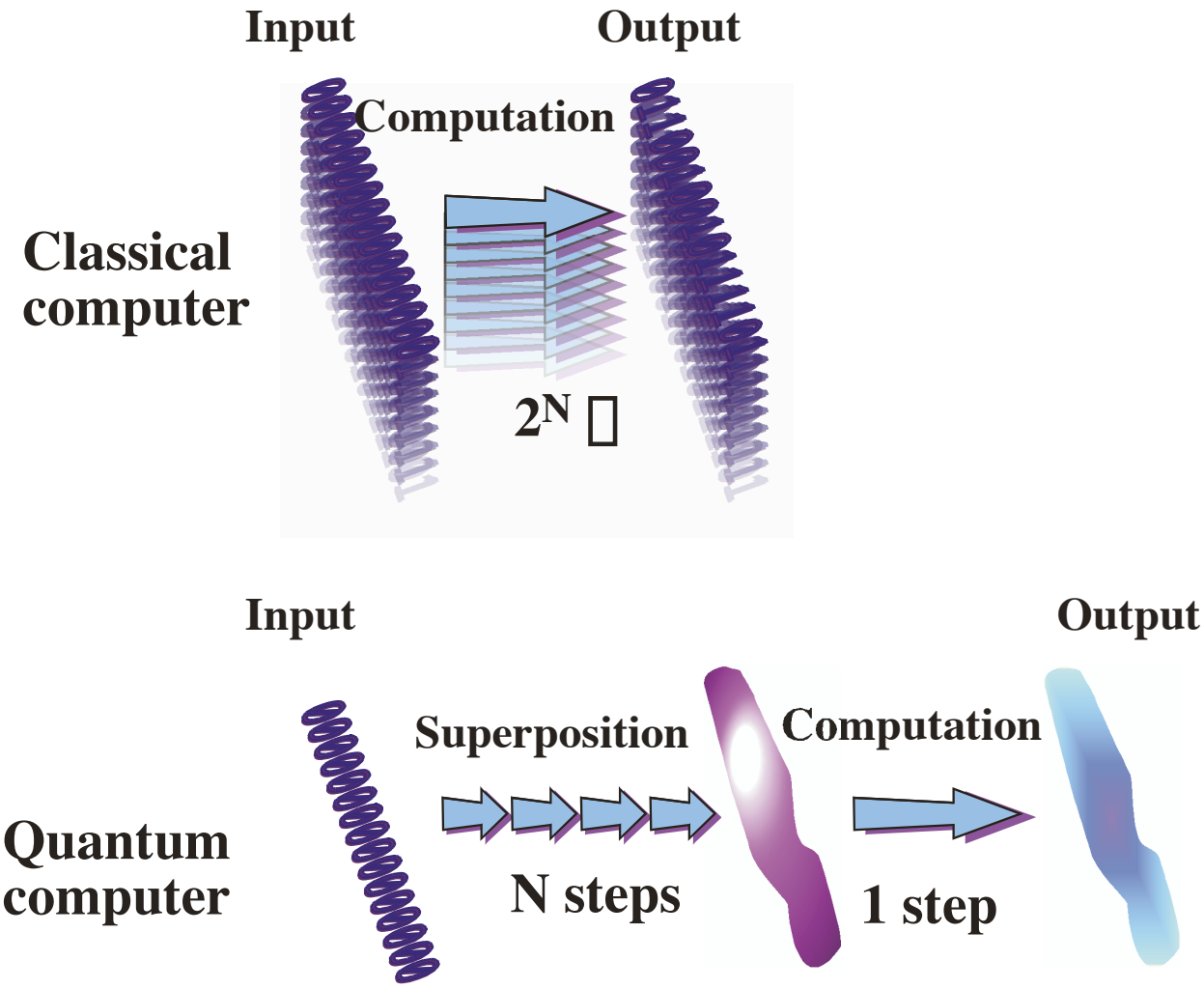
Chapter 8: Quantum Algorithms

- Why quantum?
- The Deutsch-Jozsa algorithm
- The Shor algorithm
- The Grover algorithm



Why Quantum Computing ?

Quantum computers can do anything...

...but what can they do better ?

Problems suitable for a quantum computer:

many possible states must be handled ($\rightarrow$ quantum parallelism) but

only few results are needed.

- Search in unstructured data base $\rightarrow$ Grover's search algorithm
  Quadratic speedup

- Global property of a function ("Is $f(2l+1) > 0$ for all l ?") $\rightarrow$ Shor's factoring algorithm
  Order-finding algorithm, using quantum Fourier transform. Exponential speedup

Deutsch (-Jozsa) -Algorithm

David Deutsch

More instructive than practical:
"Looking at both sides of a coin at the same time".
DEFINITION: A bit-valued function

$$f: \quad x \longrightarrow \{0,1\}$$

Richard Jozsa

is balanced, if the values 0 and 1 occur with equal frequency. In contrast, f is constant, if one of the two values does not occur at all. If f is with certainty either balanced or constant the DJ algorithm can decide between the two cases with only one evaluation of f.

Evaluation of functions with quantum gates?

$$\mathbf{U}_f|x\rangle = |f(x)\rangle$$

is impossible with unitary $\mathbf{U}_f$, because $\mathbf{U}_f^{-1}$ is undefined if f is constant, for example (irreversible). $\Rightarrow$ Keep additional information to reach reversibility, e.g. input value x.

Simplest case: The four single-qubit functions:

$$f_1 = \mathbf{1}: \quad |0\rangle \to |0\rangle, |1\rangle \to |1\rangle \qquad f_2 = \mathbf{X}: \quad |0\rangle \to |1\rangle, |1\rangle \to |0\rangle$$

are balanced,

$$f_3 = \mathbf{P}_+ + \frac{1}{\hbar}\mathbf{S}_+: \quad |0\rangle \to |0\rangle, |1\rangle \to |0\rangle \qquad f_4 = \mathbf{P}_- + \frac{1}{\hbar}\mathbf{S}_-: |0\rangle \to |1\rangle, |1\rangle \to |1\rangle$$

are constant. Unitary implementation:

$$\mathbf{U}_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

($\oplus$ is addition mod 2, or XOR). For f_1 that means

$$U_{f_1}: |0,0\rangle \to |0,0\rangle, \quad |0,1\rangle \to |0,1\rangle, \quad |1,0\rangle \to |1,1\rangle, \quad |1,1\rangle \to |1,0\rangle,$$

as a matrix in the usual basis $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} \end{pmatrix} = \mathbf{U}_{f_1}.$$

The others can be treated analoguously:

$$\mathbf{U_{f_2}} = \begin{pmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \quad \mathbf{U_{f_3}} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}, \quad \mathbf{U_{f_4}} = \begin{pmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} \end{pmatrix}.$$

All are unitary, even for constant f; at the price of a larger Hilbert space.

For unique results, initialize $y = 0$ and compute

$$\mathbf{U}_f|x, 0\rangle = |x, f(x)\rangle,$$

and obtain $f(x)$ for one input value...

Recall the Hadamard gate $\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, that means

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\ ,\ \mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad \text{both combined: } \mathbf{H}|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle).$$

(Note $|0\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.) Thus we compute both function values in one evaluation:

$$\mathbf{U}_f\mathbf{H}_x|00\rangle = \frac{1}{\sqrt{2}}\mathbf{U}_f(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle).$$

The Deutsch algorithm (1985)

$$|\psi_0\rangle = |0,1\rangle$$

$$|\psi_1\rangle = \mathbf{H}_x\mathbf{H}_y|0,1\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle\right) \otimes \left(|0\rangle - |1\rangle\right) = \frac{1}{2}\left(|00\rangle + |10\rangle - |01\rangle - |11\rangle\right)$$

Function evaluation:

$$|\psi_2\rangle = \mathbf{U}_f|\psi_1\rangle = \frac{1}{2}(|0, f(0)\rangle + |1, f(1)\rangle - |0, 1 \oplus f(0)\rangle - |1, 1 \oplus f(1)\rangle)$$

Two possible cases:

i) f constant $\Rightarrow f(1) = f(0) \Rightarrow |\psi_2\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle\right)\ \left(|f(0)\rangle - |1 \oplus f(0)\rangle\right)$.

ii) f balanced $\Rightarrow f(1) = 1 \oplus f(0) \neq f(0) \Rightarrow |\psi_2\rangle = \frac{1}{2}\left(|0\rangle - |1\rangle\right)\ \left(|f(0)\rangle - |1 \oplus f(0)\rangle\right)$

The result is encoded in the relative sign of qubit 1; it can be read out as follows:

$$|\psi_3\rangle = \mathbf{H}_x|\psi_2\rangle = |f(0) \oplus f(1)\rangle \left(\frac{|f(0)\rangle - |1 \oplus f(0)\rangle}{\sqrt{2}}\right).$$

$f(0) \oplus f(1)$ is 1 if f is balanced and 0 if f is constant.

Deutsch-Jozsa with n qubits (1992)

...is a step towards the quantum Fourier transform (QFT).

Question: Is $f(x_1, ..., x_n) = \begin{cases} 0 \\ 1 \end{cases}$ balanced or constant?

Initial state

$$|\psi_0\rangle = |\vec{0}, 1\rangle = |0\rangle_1 |0\rangle_2 \cdots |0\rangle_{n-1} |0\rangle_n |1\rangle_{n+1}.$$

is subjected to the $(n+1)$-qubit Hadamard transform

$$\mathbf{H}_{\vec{x}} \mathbf{H}_y = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes ..... \otimes \mathbf{H}_n \otimes \mathbf{H}_{n+1}$$

with the result

$$|\psi_1\rangle = \mathbf{H}_{\vec{x}} \mathbf{H}_y |\vec{0}, 1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)_1 \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)_2 \cdots \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)_n \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)_{n+1}$$

$$= \frac{1}{\sqrt{2^{(n+1)}}} \sum_{\vec{x}} |\vec{x}\rangle \left(|0\rangle - |1\rangle\right)_{n+1}.$$

The Hadamard transform thus yields a superposition of **all** 2^n possible states $|\vec{x}\rangle$ with equal phases by $\mathcal{O}(n)$ operations (highly efficient). Generalization to different phases $\rightarrow$ QFT.

The function f is again implemented unitarily and applied to the Hadamard-transfomed initial state:

$$\mathbf{U}_f|\vec{x}, y\rangle = |\vec{x}, y \oplus f(\vec{x})\rangle \qquad |\psi_2\rangle = \mathbf{U}_f|\psi_1\rangle.$$

One component of $|\psi_1\rangle$ becomes:

$$\mathbf{U}_f|\vec{x}\rangle(|0\rangle - |1\rangle) = |\vec{x}\rangle(|f(\vec{x})\rangle - |1 \oplus f(\vec{x})\rangle)$$

$$= \left\{ \begin{array}{ll} |\vec{x}\rangle(|0\rangle - |1\rangle) & \text{for } f(\vec{x}) = 0 \\ |\vec{x}\rangle(|1\rangle - |0\rangle) & \text{for } f(\vec{x}) = 1 \end{array} \right\} = (-1)^{f(\vec{x})}|\vec{x}\rangle(|0\rangle - |1\rangle).$$

Note the "typically quantum mechanical" encoding of the information as a phase. We thus obtain

$$|\psi_2\rangle = \sum_{\vec{x}} (-1)^{f(\vec{x})} \frac{|\vec{x}\rangle}{\sqrt{2^n}} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right).$$

Finally an (inverse) Hadamard transformation is applied; for that transformation we need

$$\mathbf{H}|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle) = \frac{1}{\sqrt{2}} \sum_z (-1)^{xz}|z\rangle$$

for a single qubit; for n qubits this reads

$$\mathbf{H}_{\vec{x}}|\vec{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{z}} (-1)^{\vec{x}\cdot\vec{z}}|\vec{z}\rangle \quad (\vec{x} \cdot \vec{z} = \sum_i x_i z_i).$$

the final result is thus

$$|\psi_3\rangle = \mathbf{H}_{\vec{x}}|\psi_2\rangle = \frac{1}{2^n}\sum_{\vec{z}}\sum_{\vec{x}}(-1)^{\vec{x}\cdot\vec{z}+f(\vec{x})}|\vec{z}\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

The desired result is in the amplitude of the component with $|\vec{z}\rangle = |\vec{0}\rangle$:

$$2^{-n}\sum_{\vec{x}}(-1)^{f(\vec{x})} = \begin{cases} \pm 1 \text{ if } f \text{ is constant} \\ 0 \text{ if } f \text{ is balanced} \end{cases}.$$

If f, as promised, is either balanced or constant, the result can be obtained by a single measurement. If not the amplitude will have some value between -1 and $+1$, which cannot be determined in a single measurement.

Advantage of the DJ algorithm: one evaluation of f vs. $\mathcal{O}(2^n)$ evaluations.

The additional qubit $|y\rangle$ (unchanged in the end) can be eliminated.

- Quantum parallelism: The Deutsch algorithm
- Quantum Fourier transform: Prime numbers and the Shor algorithm
- Growing and harvesting solutions: The Grover algorithm

Spies...

- want to break codes
- want to make unbreakable codes.

Quantum mechanics serves both purposes! (Shor, BB84)

Public key cryptography: everybody can encrypt, only the designated person can decrypt. Public key systems rely on number theoretic functions :

$$f_a(x_i) = y_i; \quad x_i \in \text{message text}, y_i \in \text{encrypted text}, a \text{ integer; the key.}$$

The inverse function

$$\tilde{f}_{(p,q)}(y_i) = x_i, \quad pq = a$$

depends on the prime factors p and q of a which are not easy to find.
(Find the prime factors of 29083 or 137703491 without electronic devices!)

Best known of these methods: *RSA* (Ron Rivest, Adi Shamir, Leonard Adleman). No absolute security, but all known classical factorizing algorithms for an N-digit number are exponentially expensive:

$$\#\text{ steps} \sim \exp\left(cN^{1/3}(\log N)^{2/3}\right),$$

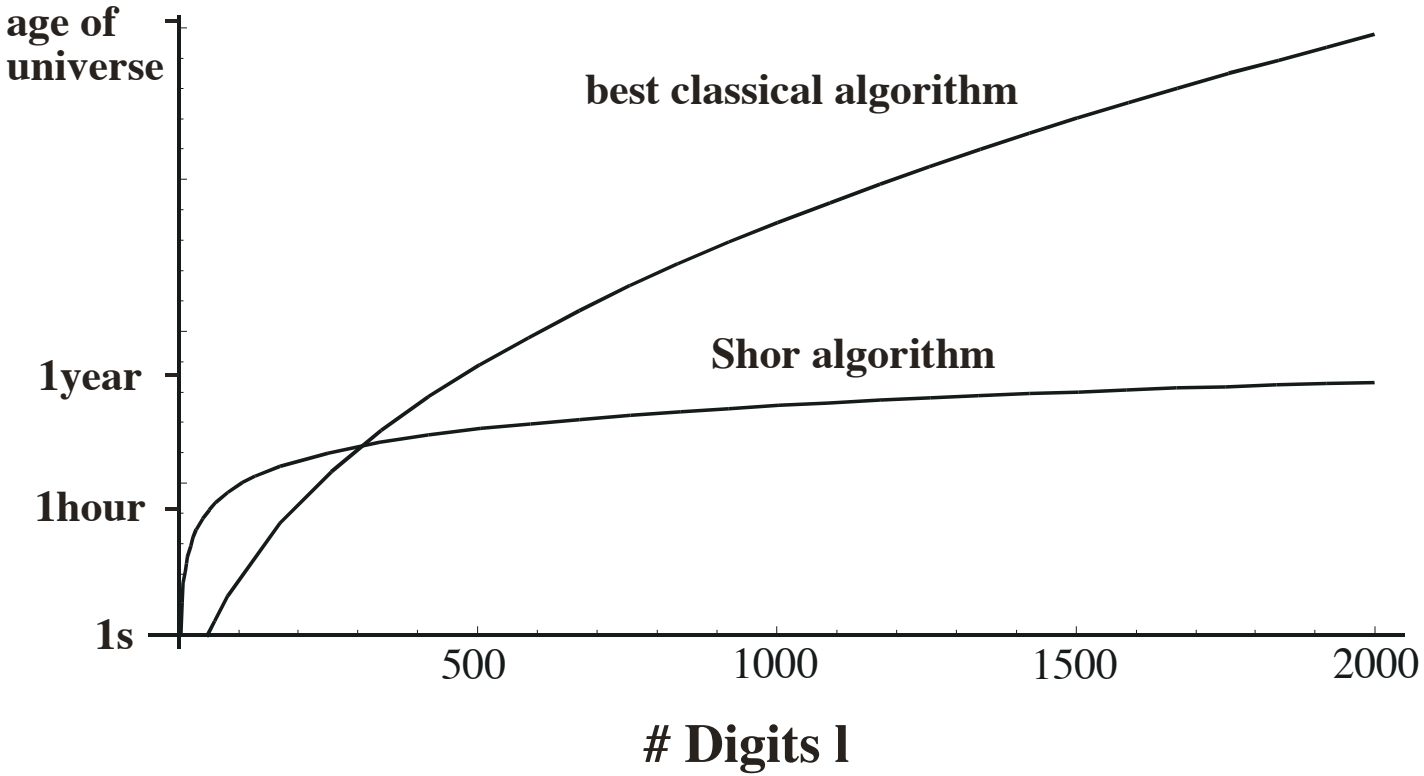
in contrast, the quantum algorithm by Peter Shor is polynomial:

$$\#\text{ steps} \sim N^2(\log N)(\log\log N)$$

1994 Peter Shor

P. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, in *35th Annual Symposium on Foundations of Computer Science*, IEEE Press, Piscataway, NJ (1994).

With his factoring algorithm, the computation time grows only algebraically, rather than exponentially with the number of digits.



The two central elements of Shor's algorithm:

Number theory
→ factorization through *period-finding* of a function
Quantum parallelism
→ efficient period-finding through *quantum Fourier transformation* (QFT, faster than FFT)

Left: Assume that 50 digits need 1 second to factorize classically, and 1 hour quantum mechanically; ⇒ for 300 digits both methods need 2.5 days,...

The Shor algorithm

Problem: to find a prime factor of a large odd number N.

- Pick a number a coprime to N, that is, a and N do not share any prime factors, $\gcd(N, a) = 1$. (If $\gcd(N, a) > 1$ you have already solved the problem!)

- Consider the modular exponential function, which is periodic with period r

$$F_N(x) = a^x \bmod N; \qquad F_N(x) = F_N(x + r), \quad r \leq N$$

  r is known as the order of $a \bmod N$.

- Three cases may arise:

  1) r is odd,
  2) r is even and $a^{r/2} \bmod N = N - 1$,
  3) r is even and $a^{r/2} \bmod N \neq N - 1$.
     Then at least one of the two numbers $\gcd(N, a^{r/2} \pm 1)$ is a nontrivial factor of N.

- For a picked at random the probability for case 3) above is greater than $\frac{3}{4}$. In other words, if we try m numbers a the chance of not finding a prime factor of N is at most 4^{-m}.

Detailed proofs involve a little number theory, see e.g. Ekert and Jozsa, Rev. Mod. Phys. **68**, 733 (1996).

The centerpiece: Calculating the order

Order $r =$ period of $F_N(x) = a^x \bmod N$. Strategy for calculating r: calculate $F_N(x)$ for many values of x in parallel and use Fourier techniques to detect the period in the sequence of function values.

For a given N two quantum registers are needed:

- a source register with K qubits such that $N^2 \leq Q := 2^K \leq 2N^2$ and
- a target register with N or more basis states, that is, at least $\log_2 N$ qubits.

Steps of the algorithm

Step 1: Initialization of both registers

$$|\psi_1\rangle = |\vec{0}\rangle|\vec{0}\rangle.$$

Step 2: Quantum Fourier transformation of the source register.
Quantum Fourier transformation = discrete Fourier transformation of an input data vector of length Q (details → later). Corresponding unitary operator on the source Hilbert space:

$$\mathbf{U}_{F_Q} : |q\rangle \mapsto \frac{1}{\sqrt{Q}} \sum_{q'=0}^{Q-1} \exp\left(i2\pi \frac{q'q}{Q}\right) |q'\rangle; \quad 0 \le q \le Q-1; \quad q = \sum_{j=0}^{K-1} q_j 2^j$$

Here $|q\rangle := |q_{K-1} \dots q_1 q_0\rangle$. The target register is not modified, so the state after step 2 is

$$|\psi_2\rangle = (\mathbf{U}_{F_Q} \otimes \mathbf{1})|\psi_1\rangle = Q^{-1/2} \sum_{q=0}^{Q-1} |q\rangle|\vec{0}\rangle;$$

All Fourier phase factors are 1 since all source qubits were initially zero. This uniform superposition can also be generated by a Hadamard transform of the source register.

Step 3 Apply the unitary gate $\mathbf{U}_a$ for modular exponentiation $q \mapsto f(q) = a^q \bmod N$

$$|\psi_3\rangle = \mathbf{U}_a|\psi_2\rangle = Q^{-1/2}\sum_{q=0}^{Q-1}|q\rangle|a^q \bmod N\rangle.$$

$Q > N^2$ function values of the function $F_N(q)$ are computed in parallel in one step, and since $r < N$ the period r must show up somewhere in this sequence of function values.
(Implementation of the modular exponential $\rightarrow$ efficient computation of (high) powers x^p of some integer x. Generate the $M+1$ numbers $x, x^2, x^4, \ldots, x^{2^M}$ by M integer multiplications. Using the binary expansion of p, x^p then can be computed using only of the order of $\log_2 p$ multiplications.)

Step 4: Apply the quantum Fourier transform again to the source register. This leads to

$$|\psi_4\rangle = (\mathbf{U}_{\mathbf{F_Q}} \otimes \mathbf{1})|\psi_3\rangle = Q^{-1}\sum_{q=0}^{Q-1}\sum_{q'=0}^{Q-1} e^{i2\pi\frac{qq'}{Q}}|q'\rangle|a^q \bmod N\rangle := \sum_{q=0}^{Q-1}\sum_{q'=0}^{Q-1}\alpha_{qq'}|q'\rangle|f(q)\rangle.$$

where $f(q)$ is the function whose periodicity we are looking for.

Step 5: Measure the source qubits in the computational basis. The probability of finding the source register in a particular state q_0 is given by

$$P(q_0) = \langle \psi_4 | \left(|q_0\rangle\langle q_0| \otimes \mathbf{1} \right) |\psi_4\rangle = \sum_{p'=0}^{Q-1} \sum_{q'=0}^{Q-1} \alpha^*_{q_0 p'} \alpha_{q_0 q'} \langle f(p')|f(q')\rangle$$

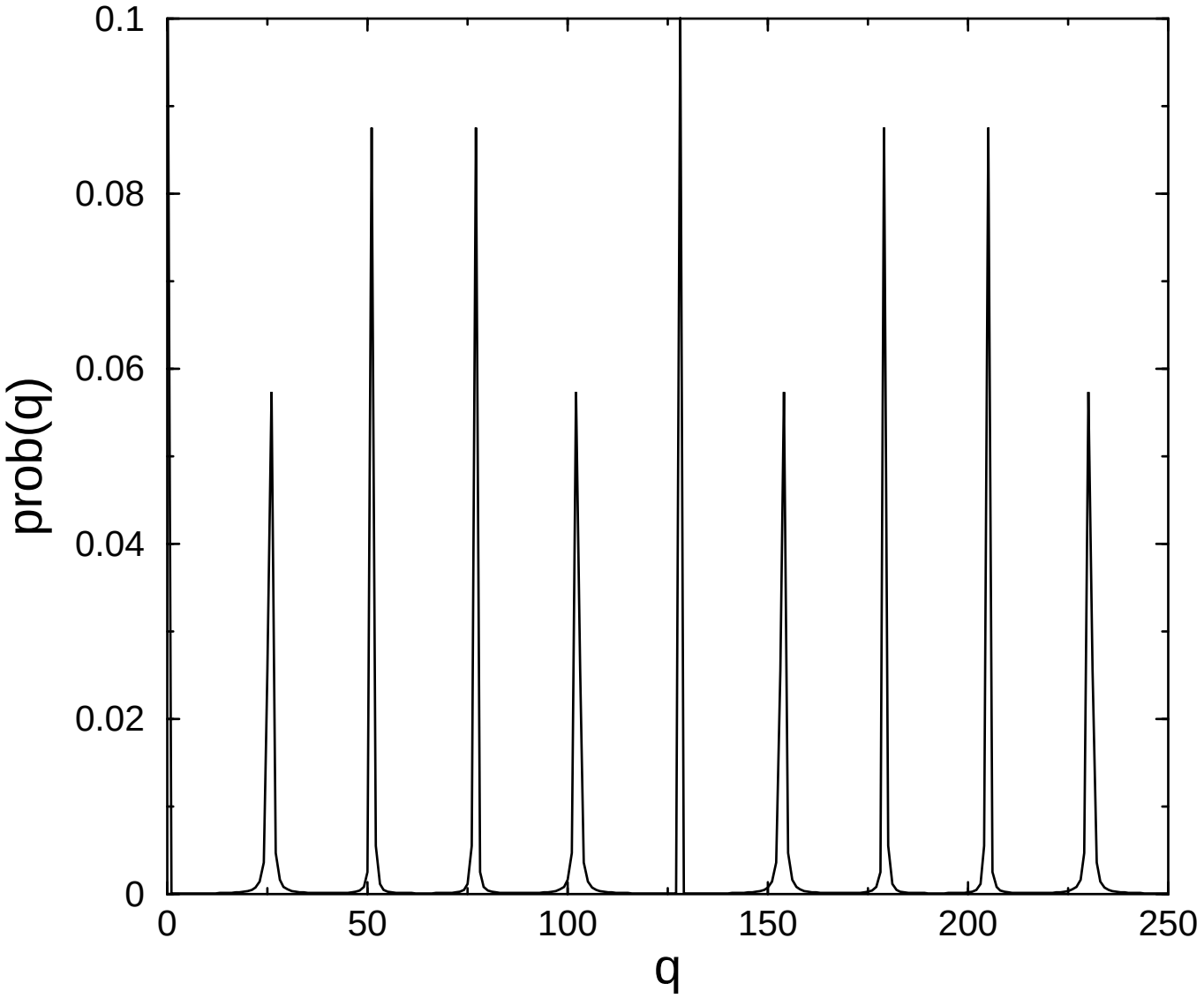
That expression displays a pattern (due to quantum interference) from the regularities of which the period r can be deduced.

The probability for finding the source register in the state $|q_0\rangle$ can be expressed in terms of a few geometrical sums:

$$P(q_0) = \frac{1}{Q^2} \sum_{j=0}^{r-1} \left| \sum_{\mu=0}^{\mathrm{int}\left(\frac{Q-1-j}{r}\right)} \left(\exp\left(i 2\pi \frac{q_0 r}{Q} \right) \right)^{\mu} \right|^2 ,$$

where "int" denotes the integer part of a real number.

The function $P(q_0)$ is shown below for $Q = 256$ and $r = 10$. From the regularities of the peak structures the period r can be deduced with a high probability (but not with certainty) if the positions of a sufficiently large number of peaks are taken into account. The full discussion of the technical details requires some additional mathematical tools.



...and finally: the quantum Fourier transform

The classical discrete Fourier transform maps a complex input vector with components $x_0, x_1, \ldots, x_{N-1}$ to the output vector (the Fourier coefficients) $y_0, y_1, \ldots, y_{N-1}$:

$$y_k = N^{-\frac{1}{2}} \sum_{j=0}^{N-1} \exp\left(\frac{i2\pi}{N} kj\right) x_j \quad \Longleftrightarrow \quad x_k = N^{-\frac{1}{2}} \sum_{j=0}^{N-1} \exp\left(-\frac{i2\pi}{N} kj\right) y_j.$$

The evaluation of the Fourier transform involves roughly N^2 complex multiplications.

The fast Fourier transform FFT (Gauß 1805) rests on the observation that by separating even and odd j one obtains (for N even)

$$y_k = N^{-\frac{1}{2}} \left[\sum_{l=0}^{\frac{N}{2}-1} \exp\left(\frac{i2\pi}{N/2} kl\right) x_{2l} + \exp\left(\frac{i2\pi}{N} k\right) \sum_{l=0}^{\frac{N}{2}-1} \exp\left(\frac{i2\pi}{N/2} kl\right) x_{2l+1} \right]$$

Both sums are again discrete Fourier transforms of $\frac{N}{2}$ data each, leading to an operation count of $2\left(\frac{N}{2}\right)^2 = \frac{1}{2}N^2$. The operation count thus has been cut in half by a simple reorganisation of the Fourier sum. Continuation of this process for $N = 2^n$ yields the FFT algorithm, with an operation count of $\mathcal{O}(N \log N)$.

The quantum Fourier transform (QFT) is an operator defined by a mapping of the basis states of an N-dimensional Hilbert space:

$$|j\rangle \mapsto N^{-\frac{1}{2}} \sum_{k=0}^{N-1} \exp\left(\frac{i2\pi}{N} jk\right) |k\rangle.$$

An arbitrary quantum state with amplitudes x_j is then mapped as

$$\sum_{j=0}^{N-1} x_j |j\rangle \mapsto \sum_{k=0}^{N-1} y_k |k\rangle$$

with $x_j \mapsto y_k$ given by the "classical" Fourier transform formula above.

For $N = 2^n$ the basis states $\{|0\rangle \ldots |2^n - 1\rangle\}$ form the computational basis for a n-qubit quantum computer. We will denote these basis states either by the integer j, or by the sequence $j_1 j_2 \ldots j_n$ from the binary representation of j

$$j = j_1 2^{n-1} + \cdots + j_n 2^0 = \sum_{\nu=1}^{n} j_\nu 2^{n-\nu}.$$

Fractional numbers (between 0 and 1) can be written as binary fractions

$$0 \,.\, j_l j_{l+1} \ldots j_m = j_l 2^{-1} + j_{l+1} 2^{-2} + \cdots + j_m 2^{-m+l-1}$$

We take another look at the quantum Fourier transform

$$|j\rangle \mapsto 2^{-\frac{n}{2}} \sum_{k=0}^{2^n-1} \exp\left(\frac{i2\pi}{2^n} jk\right) |k\rangle,$$

and insert the binary expansion of k, which leads to

$$\begin{aligned}
|j\rangle &\mapsto 2^{-\frac{n}{2}} \sum_{k_1=0}^{1} \dots \sum_{k_n=0}^{1} \exp\left(\frac{i2\pi}{2^n} j \left(\sum_{l=1}^{n} k_l 2^{n-l}\right)\right) |k_1 \dots k_n\rangle \\
&= 2^{-\frac{n}{2}} \sum_{k_1=0}^{1} \dots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} \exp(i2\pi j k_l 2^{-l}) |k_l\rangle = 2^{-\frac{n}{2}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} \exp(i2\pi j k_l 2^{-l}) |k_l\rangle\right] \\
&= 2^{-\frac{n}{2}} \bigotimes_{l=1}^{n} \left[|0\rangle_l + \exp(i2\pi j 2^{-l}) |1\rangle_l\right].
\end{aligned}$$

In the first step $|k_1 \dots k_n\rangle$ has been decomposed into an explicit tensor product $\bigotimes_{l=1}^{n} |k_l\rangle$, and the following step is just $\sum_i \sum_j a_i b_j = (\sum_i a_i)(\sum_j b_j)$.

The exponent contains a binary fraction

$$j2^{-l} = \sum_{\nu=1}^{n} j_\nu 2^{n-\nu-l} = j_1 j_2 \cdots j_{n-l} \cdot j_{n-l+1} \cdots j_n .$$

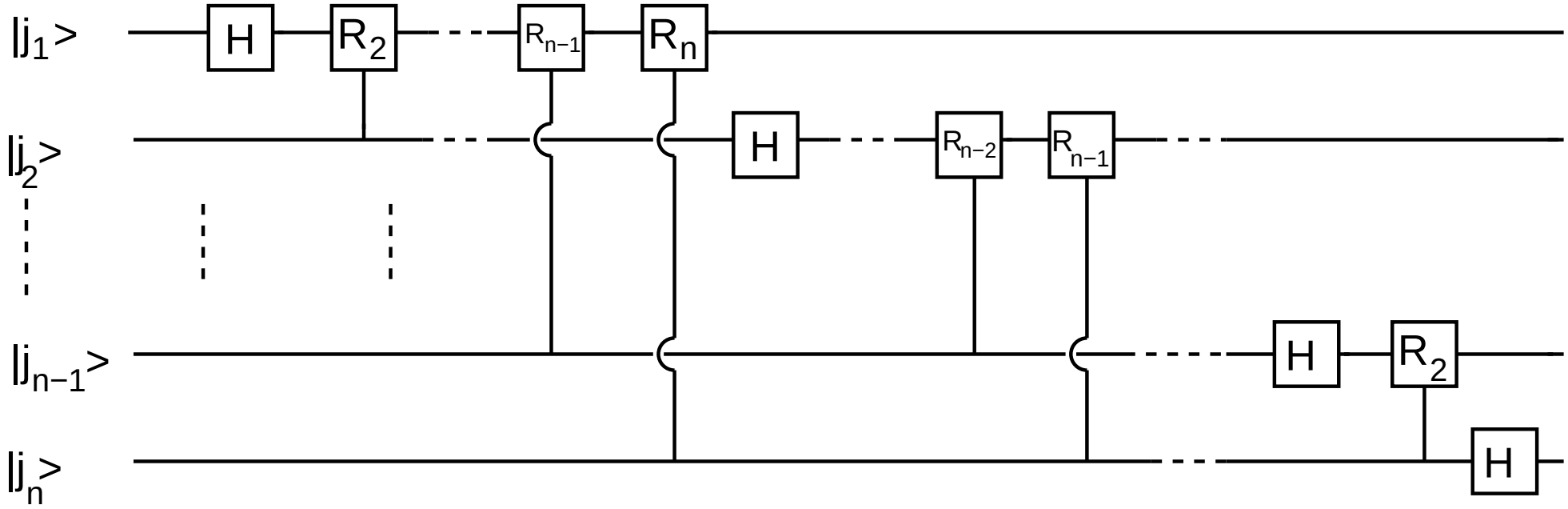
The integer part (left of the decimal point) is irrelevant because $e^{i2\pi k} = 1$ and we can write the quantum Fourier transform as

$$|j\rangle \mapsto 2^{-\frac{n}{2}} \left(|0\rangle_1 + e^{i2\pi\, 0\,.\, j_n}|1\rangle_1\right) \left(|0\rangle_2 + e^{i2\pi\, 0\,.\, j_{n-1} j_n}|1\rangle_2\right) \cdots$$
$$\cdots \left(|0\rangle_n + e^{i2\pi\, 0\,.\, j_1 j_2 \ldots j_n}|1\rangle_n\right).$$

The quantum Fourier transform is thus a simple qubit-wise phase shift: the $|1\rangle$ state of each of the n qubits is given an extra phase factor. That operation can be performed efficiently by simple quantum gates. The phase shift gate

$$\mathbf{R}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi 2^{-k}} \end{pmatrix}$$

can be converted to a controlled-$\mathbf{R}_k$ gate which applies $\mathbf{R}_k$ to the target qubit if the control qubit is in state $|1\rangle$. The controlled-$\mathbf{R}_k$ gate (for various k values) and the Hadamard gate are sufficient for the quantum Fourier transform, which is performed by the circuit shown below.



The quantum Fourier transform needs of the order of n^2 gates (operations) to Fourier transform 2^n input data. This is much better than even the FFT which needs $\mathcal{O}(n2^n)$ steps, as discussed above. Note, however, that it is not possible to get out all of the amplitudes of the final state of the quantum Fourier transform, nor is it possible to efficiently prepare the input state for arbitrary amplitudes. This restricts application of the QFT to a special class of applications, such as the Shor algorithm, which start with a fairly homogeneous input state.

- Quantum parallelism: The Deutsch algorithm
- Quantum Fourier transform: Prime numbers and the Shor algorithm
- Growing and harvesting solutions: The Grover algorithm

The Grover algorithm: Looking for a needle in a haystack

Grover's 1996 algorithm is useful for a search in an unstructured database. Note that every database is unstructured if you ask the "wrong" question. Name of a person given the street address $\rightarrow \mathcal{O}(N)$ queries in a phone directory with N entries. Grover's algorithm reduces the number of queries to $\mathcal{O}(\sqrt{N})$, which is a significant reduction for large N.

This beautiful algorithm allows the solution to "grow" out of the noise by iterating a simple procedure. As with all growing things, however, it is important to do the harvesting at the right time. It turns out that the same procedures can be used to grow the solution and to determine the time for the harvest.

Oracle functions

Consider a search space with $N = 2^n$ elements, indexed 0 to $N - 1$. Let the search problem have M solutions (persons living at the given street address), characterized by a function f with

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is a solution} \\ 0 & \text{if } x \text{ is not a solution.} \end{cases}$$

Grover's algorithm minimizes the number of calls to this "detector" function, or oracle function, as it is commonly called. The oracle function is a unitary operator $\mathbf{O}$ acting on the tensor product of the quantum register holding the index x and a single oracle qubit $|q\rangle$:

$$\mathbf{O}|x\rangle|q\rangle = |x\rangle|q \oplus f(x)\rangle.$$

The oracle qubit is flipped if x is a solution of the search problem. Initialize the oracle qubit:

$$|q_0\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \Longrightarrow \quad \mathbf{O}|x\rangle|q_0\rangle = (-1)^{f(x)}|x\rangle|q_0\rangle.$$

The oracle qubit is not changed, and will never be, $\rightarrow$ omit it for simplicity.

$$\mathbf{O}|x\rangle = (-1)^{f(x)}|x\rangle$$

The oracle marks the solutions of the search problem by a minus sign.

The search algorithm

Use the phase factors (minus signs) marking the solutions to let the amplitudes of the solution states grow out of the set of all possible states, and to "harvest" them at the right time.

Step 1. Initialize the n-qubit index register (All n qubits are set to their $|0\rangle$ states.)

$$|\psi_1\rangle = |\vec{0}\rangle$$

Step 2. Apply the Hadamard transform to generate a uniform superposition of all computational basis states.

$$|\psi_2\rangle = \mathbf{H}^{\otimes n}|\vec{0}\rangle = N^{-1/2}\sum_{x=0}^{N-1}|x\rangle \quad (N = 2^n)$$

Steps 3 and following. Iterate with the Grover operator $\mathbf{G}$

$$|\psi_{k+1}\rangle = \mathbf{G}|\psi_k\rangle$$

where the Grover operator consists of four substeps:

The Grover operator

Substep 1. Apply the oracle

$$|\psi_{k+1/4}\rangle = \mathbf{O}|\psi_k\rangle$$

Substep 2. Apply the Hadamard transform

$$|\psi_{k+1/2}\rangle = \mathbf{H}^{\otimes n}|\psi_{k+1/4}\rangle.$$

Substep 3. Apply a conditional π phase shift, that is, reverse the signs of all computational basis states except $|\vec{0}\rangle$:

$$\mathbf{C}_\pi|x\rangle = (-1)^{\delta_{x0}-1}|x\rangle; \qquad |\psi_{k+3/4}\rangle = \mathbf{C}_\pi|\psi_{k+1/2}\rangle.$$

Substep 4. Apply the Hadamard transform again

$$|\psi_{k+1}\rangle = \mathbf{H}^{\otimes n}|\psi_{k+3/4}\rangle.$$

Substeps 2,3, and 4 can be efficiently implemented on a quantum computer. The oracle *may* be computationally expensive, but we use it only once per iteration step.

The Grover operator

Substep 1. Apply the oracle

$$|\psi_{k+1/4}\rangle = \mathbf{O}|\psi_k\rangle$$

Substep 2. Apply the Hadamard transform

$$|\psi_{k+1/2}\rangle = \mathbf{H}^{\otimes n}|\psi_{k+1/4}\rangle.$$

Substep 3. Apply a conditional π phase shift, that is, reverse the signs of all computational basis states except $|\vec{0}\rangle$:

$$\mathbf{C}_\pi|x\rangle = (-1)^{\delta_{x0}-1}|x\rangle; \qquad |\psi_{k+3/4}\rangle = \mathbf{C}_\pi|\psi_{k+1/2}\rangle.$$

Substep 4. Apply the Hadamard transform again

$$|\psi_{k+1}\rangle = \mathbf{H}^{\otimes n}|\psi_{k+3/4}\rangle.$$

Substeps 2,3, and 4 can be efficiently implemented on a quantum computer. The oracle *may* be computationally expensive, but we use it only once per iteration step.

So what does this tell us?

Geometrical analysis

...of the Grover iteration step. The conditional phase shift may be written as

$$\mathbf{C}_\pi = -\mathbf{1} + 2|\vec{0}\rangle\langle\vec{0}|$$

where $\mathbf{1}$ is the n-qubit unit operator and $|\vec{0}\rangle\langle\vec{0}|$ is the projection operator onto the basis state $|\vec{0}\rangle$. We know already that

$$\mathbf{H}^{\otimes n}|\vec{0}\rangle = |\psi_2\rangle(\text{ and } \langle\psi_2| = \langle\vec{0}|\mathbf{H}^{\otimes n})$$

where $|\psi_2\rangle$ is the uniform superposition. The Grover operator thus is

$$\mathbf{G} = \mathbf{H}^{\otimes n}\mathbf{C}_\pi\mathbf{H}^{\otimes n}\mathbf{O} = (2|\psi_2\rangle\langle\psi_2| - \mathbf{1})\,\mathbf{O}.$$

The Grover iteration will turn out to be a rotation in the two-dimensional space spanned by
- the starting vector $|\psi_2\rangle$ (the uniform superposition of *all* basis states) and
- the uniform superposition of the M solutions of the search problem,

and the rotation moves the state into the right direction.

To see this we define two normalized (and mutually orthogonal) states:

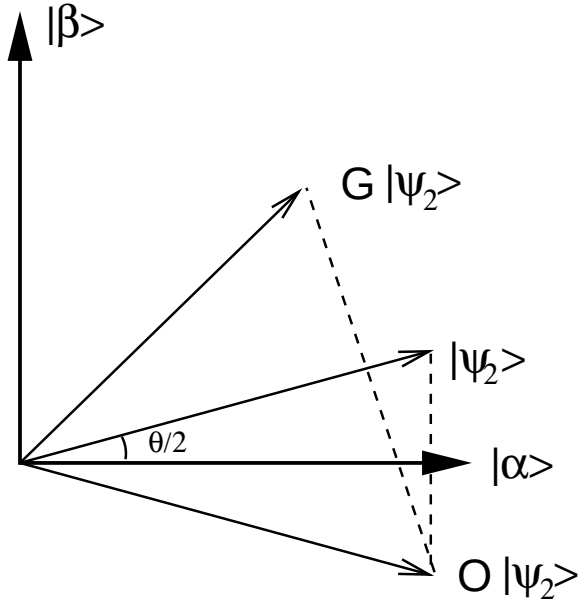
$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_x (1 - f(x))|x\rangle \ , \ |\beta\rangle = \frac{1}{\sqrt{M}} \sum_x f(x)|x\rangle$$

with the oracle function $f(x)$. $|\beta\rangle$ is the uniform superposition of the desired states and $|\alpha\rangle$ that of the remaining states. The uniform superposition of all states is $|\psi_2\rangle$ is

$$|\psi_2\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle.$$

The oracle marks solutions with a minus sign:

$$\mathbf{O}|\psi_2\rangle = \cos\frac{\theta}{2}|\alpha\rangle - \sin\frac{\theta}{2}|\beta\rangle.$$

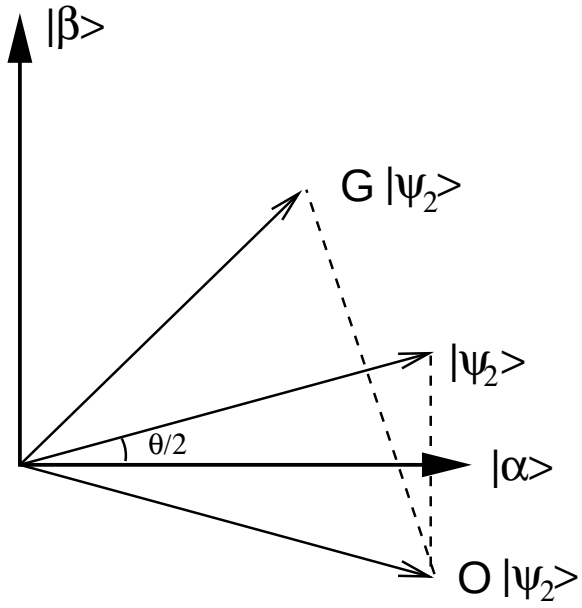


The $|\beta\rangle$ component of the initial state thus gets reversed, whereas the $|\alpha\rangle$ component remains the same: a *reflection* about the $|\alpha\rangle$ axis. The remaining three substeps of $\mathbf{G}$ are another reflection, about $|\psi_2\rangle$:

$$2|\psi_2\rangle\langle\psi_2| - \mathbf{1} = |\psi_2\rangle\langle\psi_2| - (\mathbf{1} - |\psi_2\rangle\langle\psi_2|) = \mathbf{P}_2 - \mathbf{P}_2^\perp$$

where $\mathbf{P}_2$ projects onto $|\psi_2\rangle$ and $\mathbf{P}_2^\perp$ projects onto the subspace perpendicular to $|\psi_2\rangle$.

The figure tells us that we have reached the state



$$\mathbf{G}|\psi_2\rangle = \cos\frac{3\theta}{2}|\alpha\rangle + \sin\frac{3\theta}{2}|\beta\rangle,$$

that is, $\mathbf{G}$ has performed a θ rotation. Iteration:

$$\mathbf{G}^k|\psi_2\rangle = \cos\frac{2k+1}{2}\theta|\alpha\rangle + \sin\frac{2k+1}{2}\theta|\beta\rangle.$$

Choose k such that the $|\beta\rangle$ component is as large as possible. Measurement in the computational basis will then, with high probability, produce one of the components of $|\beta\rangle$, the solutions.

From the figure and the definition of θ we see that the optimum number of iterations is the closest integer (abbreviated CI) to $\frac{\pi-\theta}{2\theta}$,

$$R := \mathrm{CI}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right) = \mathrm{CI}\left(\frac{\pi}{4\arcsin\sqrt{\frac{M}{N}}} - \frac{1}{2}\right) \leq \frac{\pi}{4}\sqrt{\frac{N}{M}}$$

since $\arcsin x > x$.

As each Grover iteration rotates the state by θ we end up at most $\theta/2$ away from $|\beta\rangle$.

For the interesting case $\frac{M}{N} \ll 1$ the error probability (given by the square of the unwanted $|\alpha\rangle$ component in the final state) is

$$p \leq \sin^2 \frac{\theta}{2} = \frac{M}{N}.$$

It is important to note that:

- iterating more than R times worsens the result,
- in this version of the algorithm, it is necessary to know M, the number of solutions.

Quantum counting

How can the number M of solutions to the search problem be counted? — Simply by a quantum algorithm involving the Grover operator $\mathbf{G}$ again: $\mathbf{G}$ is a rotation and the rotation angle is related to M. It can be determined by quantum Fourier transform techniques.

Matrix of $\mathbf{G}$ in the basis $(|\alpha\rangle, |\beta\rangle)$:

$$\mathbf{G} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The eigenvectors of this matrix are $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ \pm i \end{pmatrix}$ with eigenvalues $e^{\pm i\theta}$. Recall that $\sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$.

The problem of (approximately) counting the number M of solutions is thus reduced to estimating the phase θ of the unitary operator $\mathbf{G}$, the Grover gate. However, phase estimation is very similar to the period-finding involved in Shor's algorithm.

Phase estimation

For a given unitary operator $\mathbf{U}$ and eigenvector $|u\rangle$ we want to estimate ϕ (between 0 and 1) in:

$$\mathbf{U}|u\rangle = e^{i2\pi\phi}|u\rangle$$

Available are "black boxes" to

- prepare $|u\rangle$,
- perform controlled-$\mathbf{U}^{(2^j)}$ operations ($j = 0, 1, ..$).

Furthermore, we need two registers, one containing t qubits, initially all in the state $|0\rangle$ (t depending on the demanded accuracy and success probability of the algorithm), the second one holding the state $|u\rangle$ initially.

The algorithm:
Step 1. Apply the Hadamard transform $\mathbf{H}^{\otimes t}$ to the first register, to generate the uniform superposition state

$$\mathbf{H}^{\otimes t}|\vec{0}\rangle = \frac{1}{\sqrt{2^t}} \sum_{x=1}^{2^t} |x\rangle$$

Step 2.k ($k = 0, ..., t-1$). Apply the controlled-$\mathbf{U}^{(2^k)}$ operation to register 2, using qubit k of the first register as control qubit. This puts register 2 in state

$$|u\rangle \text{ if qubit } k \text{ is } |0\rangle \quad \text{and in state} \quad e^{i2\pi 2^k \phi}|u\rangle \text{ if qubit } k \text{ is } |1\rangle.$$

Register 2 stays in the state $|u\rangle$ all the time, up to phase factors which we can collect next to those qubits of register 1 which control them. The state of the first register thus can be written

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{i2\pi 2^{t-1}\phi}|1\rangle\right)\left(|0\rangle + e^{i2\pi 2^{t-2}\phi}|1\rangle\right)\cdots\left(|0\rangle + e^{i2\pi 2^{0}\phi}|1\rangle\right) \quad = \quad \frac{1}{2^{t/2}}\sum_{k=0}^{2^{t-1}} e^{i2\pi\phi k}|k\rangle.$$

(We have omitted the second register which remains in $|u\rangle$.) For ease of discussion, assume that ϕ is a t-bit binary fraction, $\phi = 0\,.\,\phi_1\phi_2\ldots\phi_t$ ($\phi \leq 1$). The state of register 1 is

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{i2\pi\ 0\,.\,\phi_t}|1\rangle\right)\left(|0\rangle + e^{i2\pi\ 0\,.\,\phi_{t-1}\phi_t}|1\rangle\right)\cdots\left(|0\rangle + e^{i2\pi\ 0\,.\,\phi_1\phi_2\ldots\phi_t}|1\rangle\right)$$

since $e^{i2\pi m} = 1$ for integer m.

Recall the discussion of the quantum Fourier transform from Shor's algorithm, where we constructed a circuit performing the quantum Fourier transformation

$$|j_1 \dots j_n\rangle \mapsto \frac{1}{2^{n/2}} \left(|0\rangle + e^{i2\pi\, 0\,.\, j_n}|1\rangle\right) \left(|0\rangle + e^{i2\pi\, 0\,.\, j_{n-1} j_n}|1\rangle\right) \cdots \left(|0\rangle + e^{i2\pi\, 0\,.\, j_1 j_2 \dots j_n}|1\rangle\right).$$

The inverse quantum Fourier transform can be performed by simply reversing the QFT circuit. Applying the inverse QFT to the state of register 1 leads to the state

$$|\phi_1 \dots \phi_t\rangle$$

and therefore we can measure ϕ exactly in this example, where ϕ has exactly t bits. If ϕ has more than t bits it can only be estimated. In that case, the *probability of success* of the algorithm also is an issue.

The preparation of the eigenstate $|u\rangle$ is not a problematic issue for the Grover algorithm, since the starting vector of the Grover algorithm is a combination of $|\alpha\rangle$ and $|\beta\rangle$, or equivalently, of the two eigenstates of the unitary operator $\mathbf{G}$ (the Grover operator). The phase estimation algorithm then will yield approximations to either θ or $(2\pi) - \theta$ with both of which we will be content, because knowing θ will enable us to optimize the number of iterations of $\mathbf{G}$ and therefore find a solution of the search problem with high probability.