JASS 2008

St. Petersburg, March 9th, 2008

A quantum control algorithm: Models and theory

Maximilian Fischer

Technische Universität München, Germany

Outline

Introduction

Physics

Nuclear magnetic resonance

Outline

Introduction

Physics

Nuclear magnetic resonance

Quantum computing

Computing

A quantum computer uses so called qubits instead of traditional bits to solve some problems more efficiently than on classical hardware:

- Shor's prime factorisation in polynomial time

Quantum computing

Computing

A quantum computer uses so called qubits instead of traditional bits to solve some problems more efficiently than on classical hardware:

- Shor's prime factorisation in polynomial time
- Grover-Algorithm for array searches in $\mathcal{O}(\sqrt{n})$

Quantum computing

Computing

A quantum computer uses so called qubits instead of traditional bits to solve some problems more efficiently than on classical hardware:

- Shor's prime factorisation in polynomial time
- Grover-Algorithm for array searches in $\mathcal{O}(\sqrt{n})$
- Quantum-Simulation: to simulate quantum systems, it is obviously a good choice to use quantum systems

Quantum computing

Computing

A quantum computer uses so called qubits instead of traditional bits to solve some problems more efficiently than on classical hardware:

- Shor's prime factorisation in polynomial time
- Grover-Algorithm for array searches in $\mathcal{O}(\sqrt{n})$
- Quantum-Simulation: to simulate quantum systems, it is obviously a good choice to use quantum systems

Control

Quantum control plays a key role in quantum technology, as quantum gates aren't hardwired as in traditional chips, but sophisticated manipulations of quantum systems.

What is not possible

Complexity

- BQP: The class of problems a quantum computer can solve in polynomial time with an error propability of less than $1/4$.

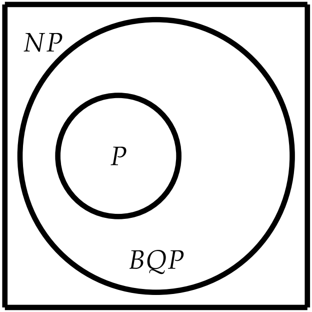


Figure: Illustration of prominent problem classes

What is not possible

Complexity

- BQP: The class of problems a quantum computer can solve in polynomial time with an error propability of less than $1/4$.
- It is known, that $P \subseteq BQP$.

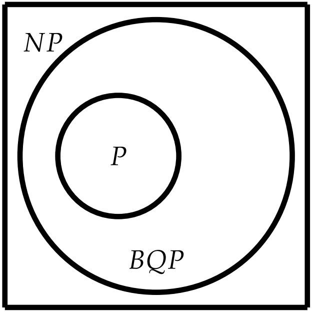


Figure: Illustration of prominent problem classes

What is not possible

Complexity

- BQP: The class of problems a quantum computer can solve in polynomial time with an error propability of less than $1/4$.
- It is known, that $P \subseteq BQP$.
- Though BQP is a subset of NP, it is not known if it is a true subset.

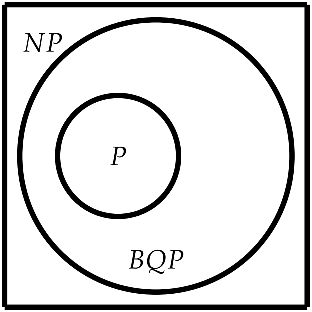


Figure: Illustration of prominent problem classes

What is not possible

Complexity

- BQP: The class of problems a quantum computer can solve in polynomial time with an error propability of less than $1/4$.
- It is known, that $P \subseteq BQP$.
- Though BQP is a subset of NP, it is not known if it is a true subset.
- Proof that $BQP \subsetneq NP$ would yield that $P \neq NP$ and therefore solve the $P = NP$ problem.

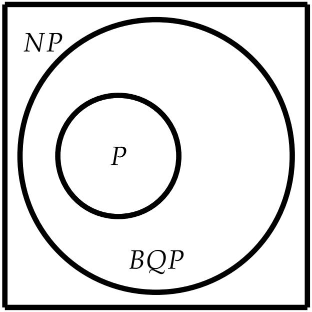


Figure: Illustration of prominent problem classes

Outline

Introduction

Physics

Nuclear magnetic resonance

Outline

Physics

- Classic mechanics
- Quantum Mechanics
- Spin
- Coupled Spins

Classic mechanics

Newton and Lagrange

- Newton's Law: $F = m \cdot \ddot{x}$

Classic mechanics

Newton and Lagrange

- Newton's Law: $F = m \cdot \ddot{x}$
- Disregarding friction, this can be shown to be equivalent to $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = 0$ with $\mathcal{L} = \frac{1}{2} \cdot m \cdot \dot{x}^2 - V(x)$ being the Lagrange-Function

Classic mechanics

Newton and Lagrange

- Newton's Law: $F = m \cdot \ddot{x}$
- Disregarding friction, this can be shown to be equivalent to $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = 0$ with $\mathcal{L} = \frac{1}{2} \cdot m \cdot \dot{x}^2 - V(x)$ being the Lagrange-Function

Hamilton

Hamilton has shown that the Lagrange equation is equivalent to this system of two partial differential equations:

- $\dot{p} = -\frac{\partial H}{\partial x}$
- $\dot{x} = \frac{\partial H}{\partial p}$

With p being the momentum $p = m \cdot \dot{x}$ and
$H = \frac{1}{2}m\dot{x}^2 + V(x) = \frac{p^2}{2m} + V(x)$ being the energy of the system.

Outline

Physics

Classic mechanics

Quantum Mechanics

Spin

Coupled Spins

Quantum Mechanics

Quantum Mechanics

The wave function

- In classical physics, $x(t)$ is a function which describes the trajectory of a mass point exactly.
- In quantum mechanics $x(t)$ is replaced by the wave function $\Psi(x,t)$.

Quantum Mechanics

The wave function

- In classical physics, $x(t)$ is a function which describes the trajectory of a mass point exactly.
- In quantum mechanics $x(t)$ is replaced by the wave function $\Psi(x, t)$.

The Correspondence principle

Classical functions become operators on the wave function whose eigenvalues are the observable values. In position space, this yields $x \longrightarrow \hat{x}$, $p \longrightarrow -i\hbar\nabla$ and $E \longrightarrow i\hbar\partial_t$.

Quantum Mechanics

The wave function

- In classical physics, $x(t)$ is a function which describes the trajectory of a mass point exactly.
- In quantum mechanics $x(t)$ is replaced by the wave function $\Psi(x,t)$.

The Correspondence principle

Classical functions become operators on the wave function whose eigenvalues are the observable values. In position space, this yields $x \to \hat{x}$, $p \to -i\hbar\nabla$ and $E \to i\hbar\partial_t$.

The Schrödinger equation

Applied to the Hamilton equation this yields the Schrödinger equation
$(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x))\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$

Outline

Physics

- Classic mechanics
- Quantum Mechanics
- Spin
- Coupled Spins

Spin

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields
- The ray got split in two parts

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields
- The ray got split in two parts

Explanation

- Electrons have an own attribute we call *spin*

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields
- The ray got split in two parts

Explanation

- Electrons have an own attribute we call *spin*
- This is correlated with a magnetic dipole moment

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields
- The ray got split in two parts

Explanation

- Electrons have an own attribute we call *spin*
- This is correlated with a magnetic dipole moment
- Spin is ***not*** angular momentum

Spin

Discovery

- In 1922, Otto Stern and Walther Gerlach experimented with accelerated atoms in inhomogenous magnetic fields
- The ray got split in two parts

Explanation

- Electrons have an own attribute we call *spin*
- This is correlated with a magnetic dipole moment
- Spin is ***not*** angular momentum
- The Schrödinger equation does not directly inhibit spin. To save us from relativistics, we apply it as a hack

The Spin Operator

The spin operator

- Let z be the distinguished axis. From the Stern-Gerlach experiment we know that the eigenvalues of $\hat{S}_z$ have to be $\pm\frac{\hbar}{2}$.

The Spin Operator

The spin operator

- Let z be the distinguished axis. From the Stern-Gerlach experiment we know that the eigenvalues of $\hat{S}_z$ have to be $\pm\frac{\hbar}{2}$.
- Hence there have to be two different linear independent eigenvectors which we call (for historical reasons) $|\uparrow\rangle$ and $|\downarrow\rangle$.

The Spin Operator

The spin operator

- Let z be the distinguished axis. From the Stern-Gerlach experiment we know that the eigenvalues of $\hat{S}_z$ have to be $\pm\frac{\hbar}{2}$.
- Hence there have to be two different linear independent eigenvectors which we call (for historical reasons) $|\uparrow\rangle$ and $|\downarrow\rangle$.
- Therefore we can write the spin state of our electron as a complex linear combination of these two vectors.

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle \in \mathbb{C}^2$$

The Spin Operator

The spin operator

- Let z be the distinguished axis. From the Stern-Gerlach experiment we know that the eigenvalues of $\hat{S}_z$ have to be $\pm\frac{\hbar}{2}$.
- Hence there have to be two different linear independent eigenvectors which we call (for historical reasons) $|\uparrow\rangle$ and $|\downarrow\rangle$.
- Therefore we can write the spin state of our electron as a complex linear combination of these two vectors.

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle \in \mathbb{C}^2$$

- Because $|\alpha|^2$ equals the propability of finding $|\uparrow\rangle$ in an experiment and $|\beta|^2$ equals the propability of finding $|\downarrow\rangle$, the normation condition is $|\alpha|^2 + |\beta|^2 = 1$.

The Pauli spin matrices

From vector to matrix

- In analogy to classic angular momentum, the spin operator has to satisfy $[\hat{S}_x, \hat{S}_y] = i\hbar\hat{S}_z$ and cyclical with $[A, B] := AB - BA$ being the commutator.

The Pauli spin matrices

From vector to matrix

- In analogy to classic angular momentum, the spin operator has to satisfy $[\hat{S}_x, \hat{S}_y] = i\hbar\hat{S}_z$ and cyclical with $[A, B] := AB - BA$ being the commutator.
- The spin operators in the three dimensions can be written as matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

With $\hat{S}_i = \frac{\hbar}{2}\sigma_i$

The Pauli spin matrices

From vector to matrix

- In analogy to classic angular momentum, the spin operator has to satisfy $[\hat{S}_x, \hat{S}_y] = i\hbar\hat{S}_z$ and cyclical with $[A, B] := AB - BA$ being the commutator.
- The spin operators in the three dimensions can be written as matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

  With $\hat{S}_i = \frac{\hbar}{2}\sigma_i$

- We can test our commutator relation from above:

$$[\hat{S}_x, \hat{S}_y] = \frac{\hbar^2}{4}\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} - \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right)$$

Outline

Physics

Classic mechanics
Quantum Mechanics
Spin
Coupled Spins

Coupled systems

The Kronecker tensorproduct

- In order to couple two spins in one system, one has to calculate the kronecker product of these two systems. Therefore we yield $2^2 = 4$ new basis vectors:

$$|\uparrow\rangle \otimes |\uparrow\rangle =: |\uparrow\uparrow\rangle \tag{1}$$

$$|\uparrow\rangle \otimes |\downarrow\rangle =: |\uparrow\downarrow\rangle \tag{2}$$

$$|\downarrow\rangle \otimes |\uparrow\rangle =: |\downarrow\uparrow\rangle \tag{3}$$

$$|\downarrow\rangle \otimes |\downarrow\rangle =: |\downarrow\downarrow\rangle \tag{4}$$

Coupled systems

The Kronecker tensorproduct

- In order to couple two spins in one system, one has to calculate the kronecker product of these two systems. Therefore we yield $2^2 = 4$ new basis vectors:

$$|\uparrow\rangle \otimes |\uparrow\rangle =: |\uparrow\uparrow\rangle \tag{1}$$
$$|\uparrow\rangle \otimes |\downarrow\rangle =: |\uparrow\downarrow\rangle \tag{2}$$
$$|\downarrow\rangle \otimes |\uparrow\rangle =: |\downarrow\uparrow\rangle \tag{3}$$
$$|\downarrow\rangle \otimes |\downarrow\rangle =: |\downarrow\downarrow\rangle \tag{4}$$

- In general, one can couple n spins by producing the kronecker product of all basis vectors, yielding 2^n basic states.

Potential energy of coupled spins

- The potential between two spins is direct proportional to the scalar product of the two spin operators or to be more exactly their eigenvalues:

Potential energy of coupled spins

- The potential between two spins is direct proportional to the scalar product of the two spin operators or to be more exactly their eigenvalues:
-

$$\hat{V} = \mu \hat{S}^{(1)} \circ \hat{S}^{(2)} = \mu \left(\hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)} + \frac{1}{2} \left(\hat{S}_+^{(1)} \otimes \hat{S}_-^{(2)} + \hat{S}_-^{(1)} \otimes \hat{S}_+^{(2)} \right) \right)$$

Potential energy of coupled spins

- The potential between two spins is direct proportional to the scalar product of the two spin operators or to be more exactly their eigenvalues:
-

$$\hat{V} = \mu \hat{S}^{(1)} \circ \hat{S}^{(2)} = \mu \left(\hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)} + \frac{1}{2} \left(\hat{S}_+^{(1)} \otimes \hat{S}_-^{(2)} + \hat{S}_-^{(1)} \otimes \hat{S}_+^{(2)} \right) \right)$$

- With μ being a constant and $\hat{S}_\pm = \hat{S}_x \pm i\hat{S}_y$ with the attributes

$$\hat{S}_+ \left|\uparrow\right\rangle = 0 \qquad \hat{S}_+ \left|\downarrow\right\rangle = \hbar \left|\uparrow\right\rangle \tag{5}$$

$$\hat{S}_- \left|\uparrow\right\rangle = \hbar \left|\downarrow\right\rangle \qquad \hat{S}_- \left|\downarrow\right\rangle = 0 \tag{6}$$

Potential energy of coupled spins

- The potential between two spins is direct proportional to the scalar product of the two spin operators or to be more exactly their eigenvalues:
-

$$\hat{V} = \mu \hat{S}^{(1)} \circ \hat{S}^{(2)} = \mu \left(\hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)} + \frac{1}{2} \left(\hat{S}_+^{(1)} \otimes \hat{S}_-^{(2)} + \hat{S}_-^{(1)} \otimes \hat{S}_+^{(2)} \right) \right)$$

- With μ being a constant and $\hat{S}_\pm = \hat{S}_x \pm i\hat{S}_y$ with the attributes

$$\hat{S}_+ \left|\uparrow\right\rangle = 0 \qquad \hat{S}_+ \left|\downarrow\right\rangle = \hbar \left|\uparrow\right\rangle \tag{5}$$

$$\hat{S}_- \left|\uparrow\right\rangle = \hbar \left|\downarrow\right\rangle \qquad \hat{S}_- \left|\downarrow\right\rangle = 0 \tag{6}$$

- We can describe the complete potential of a system by a hermitian $2^n \times 2^n$ matrix with vanishing trace.

Outline

Introduction

Physics

Nuclear magnetic resonance

Outline

Nuclear magnetic resonance

preposition

Some Physics

The GRAPE algorithm

Nuclear magnetic resonance

NMR

- Nuclei of atoms have their own spin

Figure: 900MHz, 21.2 T NMR Magnet at HWB-NMR, Birmingham, UK; credit: wikipedia

Nuclear magnetic resonance

NMR

- Nuclei of atoms have their own spin
- One can couple multiple spins in an experimental setup

Figure: 900MHz, 21.2 T NMR Magnet at HWB-NMR, Birmingham, UK; credit: wikipedia

Nuclear magnetic resonance

NMR

- Nuclei of atoms have their own spin
- One can couple multiple spins in an experimental setup
- Spins can be manipulated by external magnetic fields

Figure: 900MHz, 21.2 T NMR Magnet at HWB-NMR, Birmingham, UK; credit: wikipedia

Nuclear magnetic resonance

NMR

- Nuclei of atoms have their own spin
- One can couple multiple spins in an experimental setup
- Spins can be manipulated by external magnetic fields
- Spins can be measured by stimulated emission of radiaton

Figure: 900MHz, 21.2 T NMR Magnet at HWB-NMR, Birmingham, UK; credit: wikipedia

Technical challenges

- Strong magnetic fields ($\approx 20\mathrm{T}$)

Technical challenges

- Strong magnetic fields ($\approx 20\mathrm{T}$)
- Energy relaxation: The system returns to the ground state, the qubits are erased.

Technical challenges

- Strong magnetic fields ($\approx 20\mathrm{T}$)
- Energy relaxation: The system returns to the ground state, the qubits are erased.
- Decoherence: The superposition of the spins is destroyed by interaction with the environment ("‘super selection rule"’)

Outline

Nuclear magnetic resonance

preposition

Some Physics

The GRAPE algorithm

NMR and the Schrödinger equation

We remember

$$\hat{H}\Psi(x,t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$$

NMR and the Schrödinger equation

We remember

$$\hat{H}\Psi(x,t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$$

In our case

- Our particles don't move, so we can abbandon the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$.

NMR and the Schrödinger equation

We remember

$$\hat{H}\Psi(x,t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$$

In our case

- Our particles don't move, so we can abbandon the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$.
- We already know the potential for two particles. For n particles, this yields

$$\hat{V}(x) = \frac{1}{2}\sum_{i \neq j} \mu_{ij}\hat{S}^{(i)} \circ \hat{S}^{(j)}$$

NMR and the Schrödinger equation

We remember

$$\hat{H}\Psi(x,t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$$

In our case

- Our particles don't move, so we can abbandon the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$.
- We already know the potential for two particles. For n particles, this yields

$$\hat{V}(x) = \frac{1}{2}\sum_{i\neq j}\mu_{ij}\hat{S}^{(i)} \circ \hat{S}^{(j)}$$

- This is a $2^n \times 2^n$ matrix which can be diagonalised. In the following, we will refer to this diagnoalised matrix as H_d

The control term (1)

How to control our system

Previously we stated that the spin system can be controlled by external magnetic fields. In our formal model this can be read as application of the $\hat{S}_{\pm}$ operators on single spins.

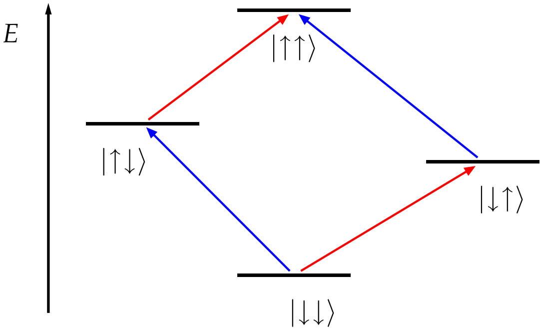


Figure: Induced spinflips in a two particle system: red is $\mathbb{1}_2 \otimes \hat{S}_+$ and blue is $\hat{S}_+ \otimes \mathbb{1}_2$.

The control term (2)

In general

For n spins which can be separatley influenced, the controlled potential is

$$\hat{V}_c = \sum_{k=0}^{n-1} (a_k \cdot \mathbb{1}_{2^k} \otimes \sigma_x \otimes \mathbb{1}_{2^{n-k-q}} + b_k \cdot \mathbb{1}_{2^k} \otimes \sigma_y \otimes \mathbb{1}_{2^{n-k-q}})$$

Which we will call H_c.

The control term (2)

In general

For n spins which can be separatley influenced, the controlled potential is

$$\hat{V}_c = \sum_{k=0}^{n-1} (a_k \cdot \mathbb{1}_{2^k} \otimes \sigma_x \otimes \mathbb{1}_{2^{n-k-q}} + b_k \cdot \mathbb{1}_{2^k} \otimes \sigma_y \otimes \mathbb{1}_{2^{n-k-q}})$$

Which we will call H_c.

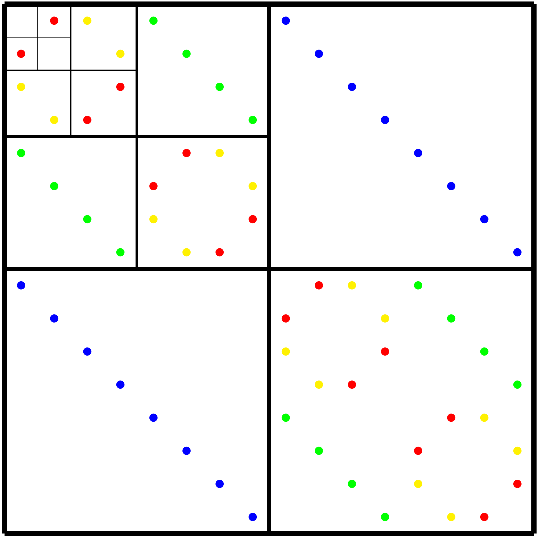
Recursion

One can build the matrix of H_c for n particles using the following recursion:

$$A_{n+1} = \begin{pmatrix} A_n & \mathbb{1}_{2^n} \\ \mathbb{1}_{2^n} & A_n \end{pmatrix}$$

With $A_0 = (0)$ being the matrix for zero particles.

Eye candy



Outline

Nuclear magnetic resonance

preposition

Some Physics

The GRAPE algorithm

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t \Psi = \hat{H}\Psi$

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t\Psi = \hat{H}\Psi$
- In the Gaussian system: $i\partial_t\Psi = \hat{H}\Psi$

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t\Psi = \hat{H}\Psi$
- In the Gaussian system: $i\partial_t\Psi = \hat{H}\Psi$
- The solution is obviously: $\Psi(t) = e^{-i\hat{H}t}\Psi(0)$

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t\Psi = \hat{H}\Psi$
- In the Gaussian system: $i\partial_t\Psi = \hat{H}\Psi$
- The solution is obviously: $\Psi(t) = e^{-i\hat{H}t}\Psi(0)$
- With the matrix exponential function $e^{\hat{H}} = \sum_{k=0}^{\infty} \frac{\hat{H}^k}{k!}$

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t\Psi = \hat{H}\Psi$
- In the Gaussian system: $i\partial_t\Psi = \hat{H}\Psi$
- The solution is obviously: $\Psi(t) = e^{-i\hat{H}t}\Psi(0)$
- With the matrix exponential function $e^{\hat{H}} = \sum_{k=0}^{\infty} \frac{\hat{H}^k}{k!}$
- Our Hamiltonian was: $\hat{H} = H_d + H_c(a_1(t), b_1(t), \ldots) = H_d + H_c(u_1(t), \ldots) = H_d + \sum_j H_j(t)$ With $H_j(t)$ piecewise constant on $t + \Delta t$

A solution for the Schrödinger equation

Forward Propagation

- The time-independent Schrödinger equation: $i\hbar\partial_t\Psi = \hat{H}\Psi$
- In the Gaussian system: $i\partial_t\Psi = \hat{H}\Psi$
- The solution is obviously: $\Psi(t) = e^{-i\hat{H}t}\Psi(0)$
- With the matrix exponential function $e^{\hat{H}} = \sum_{k=0}^{\infty} \frac{\hat{H}^k}{k!}$
- Our Hamiltonian was: $\hat{H} = H_d + H_c(a_1(t), b_1(t), \ldots) = H_d + H_c(u_1(t), \ldots) = H_d + \sum_j H_j(t)$ With $H_j(t)$ piecewise constant on $t + \Delta t$
- So in our case the solution is: $\Psi(t) = e^{-i\Delta t\hat{H}(t_k)}e^{-i\Delta t\hat{H}(t_{k-1})} \ldots e^{-i\Delta t\hat{H}(t_1)}\Psi(0) =: U(t)\Psi(0)$ With $k\Delta t = t$

Quantum gate construction

Problem description

- A quantum gate is an operation on the spin state of the system which performs a desired change in it, e.g. NOT, NAND, XOR,. . ..

Quantum gate construction

Problem description

- A quantum gate is an operation on the spin state of the system which performs a desired change in it, e.g. NOT, NAND, XOR,. . ..
- For each of these gates the desired operation can be described by a matrix U_G.

Quantum gate construction

Problem description

- A quantum gate is an operation on the spin state of the system which performs a desired change in it, e.g. NOT, NAND, XOR,. . ..
- For each of these gates the desired operation can be described by a matrix U_G.
- So the challenge is: adjusting $H_j(t_k)$ so that $U(t)$ overlaps best with U_G for a given time $t = T$.

Quantum gate construction

Problem description

- A quantum gate is an operation on the spin state of the system which performs a desired change in it, e.g. NOT, NAND, XOR,. . ..
- For each of these gates the desired operation can be described by a matrix U_G.
- So the challenge is: adjusting $H_j(t_k)$ so that $U(t)$ overlaps best with U_G for a given time $t = T$.

The GRAPE algorithm

It can be shown that maximising $\Re\, tr(U_G^\dagger U(T))$ subject to $\partial_t U(t) = -i\hat{H}U(t)$ optimizes the propagator.

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j)U(t_k)\right)\right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j) U(t_k) \right) \right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j)U(t_k) \right) \right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j) U(t_k) \right) \right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation
   $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation
   $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re\left(tr\left(\Lambda^\dagger(t_k)(-i\hat{H}_j)U(t_k)\right)\right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j)U(t_k) \right) \right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

GRAPE

1. Set initial controls $u_j^{(r)}(t_k)$ for all times t_k ($k \in \{1, 2, \ldots, M\}$) at random or by guess
2. **For each** $k \in \{1, \ldots, M\}$ **do**:
   2.1 Calculate the forward-propagation
   $U(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k-1})} \ldots e^{-i\Delta t \hat{H}(t_1)}$
   2.2 Calculate the backward-propagation
   $\Lambda(t_k) = e^{-i\Delta t \hat{H}(t_k)} e^{-i\Delta t \hat{H}(t_{k+1})} \ldots e^{-i\Delta t \hat{H}(t_M)}$
   2.3 Update $u_j^{(r+1)}(t_k) = u_j^{(r)}(t_k) + \varepsilon \Re \left(tr \left(\Lambda^\dagger(t_k)(-i\hat{H}_j) U(t_k) \right) \right)$
3. Return to step 2 with the new controls $u_j^{(r+1)}$

Challenges of GRAPE

- GRAPE converges to a local optimum of $U(t)$. It is necessary to re-run it a couple of times with different initial values to confirm that the global maximum is reached altough this cannot be proven.

Challenges of GRAPE

- GRAPE converges to a local optimum of $U(t)$. It is necessary to re-run it a couple of times with different initial values to confirm that the global maximum is reached altough this cannot be proven.
- One has to calculate the exponential of a sparse matrix $U_k := e^{-i\Delta t \hat{H}(t_k)}$

Challenges of GRAPE

- GRAPE converges to a local optimum of $U(t)$. It is necessary to re-run it a couple of times with different initial values to confirm that the global maximum is reached altough this cannot be proven.
- One has to calculate the exponential of a sparse matrix $U_k := e^{-i\Delta t \hat{H}(t_k)}$
- One has to calculate the product of many different matrices $U(t_k) = U_k \cdot U_{k-1} \cdots U_1$

Challenges of GRAPE

- GRAPE converges to a local optimum of $U(t)$. It is necessary to re-run it a couple of times with different initial values to confirm that the global maximum is reached altough this cannot be proven.
- One has to calculate the exponential of a sparse matrix $U_k := e^{-i\Delta t \hat{H}(t_k)}$
- One has to calculate the product of many different matrices $U(t_k) = U_k \cdot U_{k-1} \cdots U_1$
- One has to calculate the trace $tr\{(U_k U_{k+1} \cdots U_M)(-i\hat{H}_j)(U_k U_{k-1} \cdots U_1)\} \forall j, k$

Conclusion

- The Schrödinger equation:
  $\left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$

Conclusion

- The Schrödinger equation:
  $\left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$
- In NMR spectroscopy, the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$ can be abbandoned an $\hat{V}(x)$ splits in H_d and H_c

Conclusion

- The Schrödinger equation:
  $\left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$
- In NMR spectroscopy, the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$ can be abbandoned an $\hat{V}(x)$ splits in H_d and H_c
- H_d can be diagonalised whereas H_c has a recursive shape

Conclusion

- The Schrödinger equation:
  $\left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$
- In NMR spectroscopy, the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$ can be abbandoned an $\hat{V}(x)$ splits in H_d and H_c
- H_d can be diagonalised whereas H_c has a recursive shape
- In order to overlap the time propagation $U(t) = e^{-i\hat{H}\Delta t}$ with the desired matrix U_G, a gradient flow algorithm can be utilized

Conclusion

- The Schrödinger equation:
  $\left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}(x)\right)\Psi(x,t) = i\hbar\partial_t\Psi(x,t)$
- In NMR spectroscopy, the kinetic term $-\frac{\hbar^2}{2m}\nabla^2$ can be abbandoned an $\hat{V}(x)$ splits in H_d and H_c
- H_d can be diagonalised whereas H_c has a recursive shape
- In order to overlap the time propagation $U(t) = e^{-i\hat{H}\Delta t}$ with the desired matrix U_G, a gradient flow algorithm can be utilized
- This leads to some numerical challenges thus as calculating a matrix exponential as well as producing the product of many matrices

References

- Gradl et al., *Parallelising Matrix Operations on Clusters for an Optimal Control-Based Quantum Compiler*
- Häffner et al., *Scalable multiparticle entanglement of trapped ions*, Nature 438,643 (2005)
- Schwabl, *Quantenmechanik*, 6. Auflage, Springer, 2007
- Fließbach, *Mechanik*, 4. Auflage, Spektrum, 2003
- Fließbach, *Quantenmechanik*, 4. Auflage, Spektrum, 2005
- Waldherr, *Die Matrix-Exponentialabbildung: Eigenschaften und Algorithmen*, 2007
- Waldherr, *Numerical Linear Algebra Tasks in a Quantum Control Problem*, 2008
- Bornemann, *Quantenrechnen*, Technische Universität München, 2003

Lie algebra (1)

Definition

Let V be a vector space over a field F with a binary operation $[\cdot, \cdot]$

$$[\cdot, \cdot] \ : \ V \times V \to V$$

which satisfies the following relations:

Lie algebra (1)

Definition

Let V be a vector space over a field F with a binary operation $[\cdot,\cdot]$

$$[\cdot,\cdot] \ : \ V \times V \rightarrow V$$

which satisfies the following relations:

- Bilinearity: $[x + \lambda y, z] = [x, z] + \lambda[y, z]$ and $[x, y + \lambda z] = [x, y] + \lambda[x, z]$

Lie algebra (1)

Definition

Let V be a vector space over a field F with a binary operation $[\cdot,\cdot]$

$$[\cdot,\cdot]\ :\ V \times V \to V$$

which satisfies the following relations:

- Bilinearity: $[x+\lambda y, z] = [x,z] + \lambda[y,z]$ and $[x, y+\lambda z] = [x,y] + \lambda[x,z]$
- Skew-symmetry: $[x,x] = 0$

Lie algebra (1)

Definition

Let V be a vector space over a field F with a binary operation $[\cdot,\cdot]$

$$[\cdot,\cdot] \; : \; V \times V \to V$$

which satisfies the following relations:

- Bilinearity: $[x+\lambda y, z] = [x,z] + \lambda[y,z]$ and $[x, y+\lambda z] = [x,y] + \lambda[x,z]$
- Skew-symmetry: $[x,x] = 0$
- Jacobi Identity: $[x,[y,z]] + [y,[z,x]] + [z,[x,y]] = 0$

Lie algebra (1)

Definition

Let V be a vector space over a field F with a binary operation $[\cdot,\cdot]$

$$[\cdot,\cdot] \; : \; V \times V \rightarrow V$$

which satisfies the following relations:

- Bilinearity: $[x+\lambda y, z] = [x,z] + \lambda[y,z]$ and $[x, y+\lambda z] = [x,y] + \lambda[x,z]$
- Skew-symmetry: $[x,x] = 0$
- Jacobi Identity: $[x,[y,z]] + [y,[z,x]] + [z,[x,y]] = 0$

$\forall x,y,z \in V, \forall \lambda \in F$
Then V is a Lie algebra.

Lie algebra (2)

Examples

- The well-known $\mathbb{R}^3$ with the cross product.
- Our previously defined Pauli-Matrices.

Kronecker product (1)

Definition

Let $A \in \mathcal{C}^{m \times n}$, $B \in \mathcal{C}^{r \times s}$. Then the Kronecker product $A \otimes B \in \mathcal{C}^{mr \times ns}$ of A and B is defined as:

$$A \otimes B = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \otimes B := \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

Attributes (1)

- Bilinearity:
  - $A \otimes (B + C) = A \otimes B + A \otimes C$
  - $(A + B) \otimes C = A \otimes C + B \otimes C$
  - $\lambda(A \otimes B) = (\lambda A) \otimes B = A \otimes (\lambda B)$
- associativity: $A \otimes (B \otimes C) = (A \otimes B) \otimes C$

Kronecker product (2)

Attributes (2)

- transposition: $(A \otimes B)^T = A^T \otimes B^T$
- $\forall A, B \in \mathbb{C}^{n \times n},\ C, D \in \mathbb{C}^{m \times m}: \quad (AB) \otimes (CD) = (A \otimes C)(B \otimes D)$
- The kronecker product of diagonal matrices is a diagonal matrix
- $\mathbb{1}_{2^q} = \underbrace{\mathbb{1}_2 \otimes \cdots \otimes \mathbb{1}_2}_{\text{q times}}$
- $tr(A \otimes B) = tr(A) \cdot tr(B)$

Drift-Term

Two spin system

$$|\Psi_1\rangle := |\uparrow\uparrow\rangle \quad |\Psi_2\rangle := |\uparrow\downarrow\rangle \quad |\Psi_3\rangle := |\downarrow\uparrow\rangle \quad |\Psi_4\rangle := |\downarrow\downarrow\rangle$$

$$\hat{H}_d = \hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)} + \frac{1}{2}\left(\hat{S}_+^{(1)} \otimes \hat{S}_-^{(2)} + \hat{S}_-^{(1)} \otimes \hat{S}_+^{(2)}\right)$$

Non-diagonalised Hamiltonian for two-spin system

$$\hat{H}_d = \frac{\hbar^2}{4}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Reference: Myself, so it could be faulty.

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. If $gcd(x, n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v + r) = f(v)$ (quantum stuff here)
4. If r is odd $\rightarrow$ go back to step 1
5. If $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. If $gcd(x,n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v+r) = f(v)$ (quantum stuff here)
4. If r is odd $\rightarrow$ go back to step 1
5. If $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. If $gcd(x,n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v+r) = f(v)$ (quantum stuff here)
4. If r is odd $\rightarrow$ go back to step 1
5. If $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. **If** $gcd(x, n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v + r) = f(v)$ (quantum stuff here)
4. **If** r is odd $\rightarrow$ go back to step 1
5. **If** $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. **If** $gcd(x, n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v + r) = f(v)$ (quantum stuff here)
4. **If** r is odd $\rightarrow$ go back to step 1
5. **If** $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. **If** $gcd(x,n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v+r) = f(v)$ (quantum stuff here)
4. **If** r is odd $\rightarrow$ go back to step 1
5. **If** $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (1)

Factorize a number n

1. Pick random $1 < x < n$
2. **If** $gcd(x, n) > 1 \rightarrow$ **success**
3. Use the period-finding subroutine to find r, the period of $f(v) = x^v \mod n$ i.e. the smallest integer r for which $f(v + r) = f(v)$ (quantum stuff here)
4. **If** r is odd $\rightarrow$ go back to step 1
5. **If** $x^{\frac{r}{2}} = -1 \mod n \rightarrow$ go back to step 1
6. $gcd(x^k - 1, n)$ is a nontrivial factor of n. **success**

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_{x} |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_{x} \sum_{y} \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_x |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_x \sum_y \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_x |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_x \sum_y \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_{x} |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_{x} \sum_{y} \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_{x} |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_{x} \sum_{y} \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Shor's Algorithm (2)

Period finding subroutine

You will need at least Q qubits, where $n^2 \leq Q < 2n^2$.

1. Initialize the qubits to $Q^{-\frac{1}{2}} \sum_{x=0}^{Q-1} |x\rangle |0\rangle$
2. Construct $f(x)$ as a quantum function and apply it to the state, to obtain
$$Q^{-\frac{1}{2}} \sum_{x} |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform to get the final state
$$Q^{-1} \sum_{x} \sum_{y} \omega^{xy} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain an equally distributed multiple of $f(x)/r$.
5. Repeat a couple of times to obtain a working candidate for r

Stern Gerlach experiment