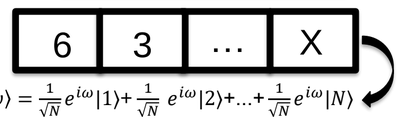
Adiabatic Oracle For Grover's Algorithm

Benjamin Freiman

Grovers Algorithm

Encoding Information

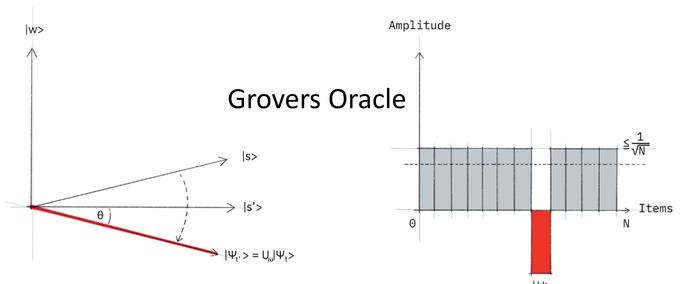


$$|\psi\rangle = \frac{1}{\sqrt{N}}e^{i\omega}|1\rangle + \frac{1}{\sqrt{N}}e^{i\omega}|2\rangle + \ldots + \frac{1}{\sqrt{N}}e^{i\omega}|N\rangle$$

Grovers Oracle

$$f(x) = \begin{cases} 1, & x = \omega \\ 0, & x \neq \omega \end{cases}$$

$$\widehat{O_f}|\psi\rangle = \sum_{m\neq\omega} c_m|m\rangle - c_\omega|\omega\rangle$$



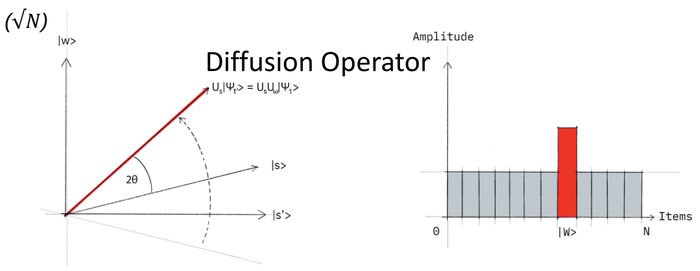
Ref [2]

Diffusion Operator

$O(\sqrt{N})$

$$\widehat{D}|\psi\rangle = H^{\otimes n}RH^{\otimes n}\widehat{O_f}|\psi\rangle$$
$$R = 2|0\rangle\langle 0| - I$$

The diffusion operator is used to reflect the states over the initial state of the system ($|0\rangle$). The Hadamard Gates allow operations on the phase of the states but also return the state to the original basis.



Adiabatic Evolution

Ref. [3]

$$H(t) = A(t)H_f + B(t)H_M$$
Where $A(t) = 0 \rightarrow 1$ and $B(t) = 1 \rightarrow 0$

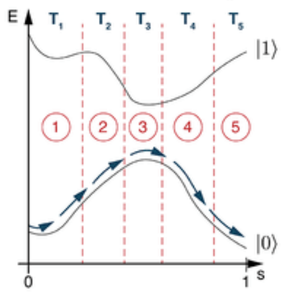
Let there be a system with instantaneous eigenstates of some Hamiltonian:

$$H(t) = \sum_n E_n(t)|n(t)\rangle\langle n(t)|$$

A requirement is that the probability of measuring a state does not change in time, however, the phases attached to each eigenstate can change:

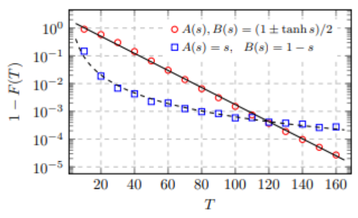
$$\theta_n(t) = \underbrace{\int_0^t E_n(\tau)d\tau}_{\text{Dynamical Phase}} - \underbrace{i\int_0^t \langle n(\tau)|\dot{n}(\tau)\rangle d\tau}_{\text{Geometrical Phase}}$$

Ensure that the ground state energy is sufficiently far apart from all other energy levels. Otherwise, there might be



Adiabatic Time Schedule to Suppress Adiabatic Errors

Ref. [1]



$$1 - F(T) \equiv 1 - |\langle\phi_T|\phi_T^{\text{id}}\rangle|^2$$

This algorithm discussed scales as O ($\log N$) on an Adiabatic Quantum Computer (AQC).

4 qbit Example

The simulation is implemented in IBM's Qiskit with two qbits of information and two ancillary qbits that are able to indicate a successful evolution. The simulation used the statevector simulator which is a circuit-based computer rather than an adiabatic AQC. The scaling of the algorithm here is much higher than implementing on an AQC. The ancillary qbits are initially in the x-basis and the second qbit is entered into a equal superposition of the eigenvectors of the z-basis. The expected outcome is the phase of the lowest eigenstate $|1\rangle$ to flip during evolution. When measuring the qbits, they are measured in the z-basis and the second qbit which was initialized as spin down $|1\rangle$ will be spin up $|0\rangle$.

1. Initialize qbits: $|\psi\rangle = |0,1,+,-\rangle$ where $|-\ +\rangle$ are eigenstates of σ_x
    1. Put the second qbit in a superposition using an H-Gate:

$$|\psi\rangle = |0,\chi,+,-\rangle \text{ where } \chi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

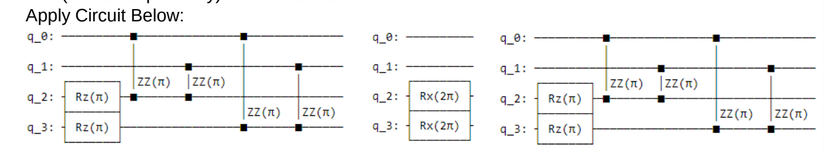
2. Evolve qbits (non-commuting operators, preform half of the first rotation operator (attached to A(t)) and the

Pseudo Code

```
Total Time = 150
Total Iterations = 1000
Time Delta = Total Time / Total Iterations

for t in range(Total Iterations):
    A = 1-t/Total Time
    B = t/Total Time
    π = (A or B respectively) π ×Time Delta
    Apply Circuit Below:
```

Note, π is simply a placeholder in the circuit visualization.



End Loop

$$R_{ZZ}(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{pmatrix}$$

$$RZ(\lambda) = \begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}$$

$$RX(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

3. Read out qbits in z-basis (For a successful evolution which happens ½ of the time, the expected result is the phase flips$-|1\rangle \rightarrow |1\rangle$)
    1. $|\psi\ (t = T)\rangle = |0,\beta,\pm,\pm\rangle$ where $\beta = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$
    2. Change Basis
        i. Apply Hadamard Gate to the second qbit

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

        ii. Apply Hadamard Gates to last qbits:

$$|\psi(t = T)\rangle = \frac{1}{\sqrt{2}}|0,0,0,0\rangle + \frac{1}{\sqrt{2}}|0,0,1,1\rangle$$

Hadamard Gate-Qiskit

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Maps the positive and negative eigenstates in the σ_x basis to the positive and negative eigenstates in the σ_z basis respectively. Because it is unitary, the H-Gate can also map the the positive and negative eigenstates in the σ_z basis to the positive and negative eigenstates in the σ_x basis as well.

Note: If the last two qbits aren't in the same state, the initial state is preserved so apply the oracle again with constant $O(1)$ overhead.

Adiabatic Oracle

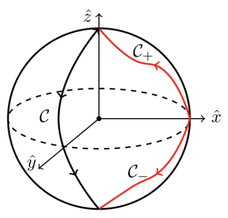
$$\widehat{O_f}|\phi\rangle = \sum_{m\neq\omega} c_m|m\rangle - c_\omega|\omega\rangle$$

H_f Identifies the elements below the threshold.

$$H(t) = A(t)H_f \otimes S_z + B(t)S_x$$
$$H(t)|\psi_m\rangle = A(t)E_mS_z + B(t)S_x$$

This oracle will flip all of the phases of eigenstates below a threshold value under the assumption that there is some Hamiltonian able to identify the elements below the threshold value.

Phase Evolution



Depending on the initial transport direction (C_+, C_-) the resultant phase offset will be $e^{-i\pi}$[1].

Initial State

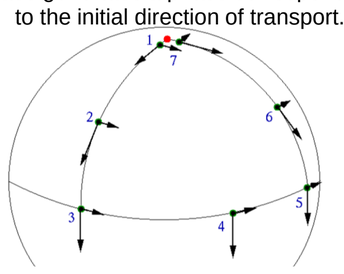
$$|\psi(t = 0)\rangle = |\phi\rangle \otimes |0_x\rangle,$$

Time Evolution

$$|\psi(t)\rangle = \sum_{\dots} c_m|m\rangle \otimes \hat{\mathcal{T}}e^{-i\int_0^t d\tau[A(\tau)E_mS_z + B(\tau)S_x]}|0_x\rangle$$

Berry Phase

Any vector which is parallel transported along a path and back to the original place generates a phase corresponding to the initial direction of transport.



Result of Berry Phase

Final State

$$|\psi(t = T)\rangle = \left(\sum_{m\neq\omega} c_m|m\rangle - c_\omega|\omega\rangle\right) \otimes |0_z\rangle.$$

Results

After running the simulation 1000 times, the successful evolutions are shown in green. The unsuccessful evolutions are in red and the unexpected errors are in white. It is evident that the number or successful runs is 479 (which is on the order of ½ of the 1000 runs). These confirm expected results [1].

State	0101	0011	0000	0110	0010	0001
Occurrence	258	240	239	259	3	1

https://github.com/bfreiman/Adiabatic-Oracle-For-Grovers-Algrithm

References

[1]B. Yan, & Nikolai A. Sinitsyn. (2023). An adiabatic oracle for Grover's algorithm. https://arxiv.org/pdf/2207.05665.pdf
[2]Grover's algorithm. IBM Quantum. (n.d.). https://quantum-computing.ibm.com/admin/docs/admin/guide/grovers-algorithm
[3]https://wdscultan.github.io/files/AE.pdf
[4]J. M Robbins and M. V. Berry. (1994) "A geometric phase for m=0 spins" J. Phys. A Math. Gen. 27 L435

Thank you to Dr. B. Yan and Dr. N. Sinitsyn at Los Alamos National Laboratory for mentoring me through this project.