

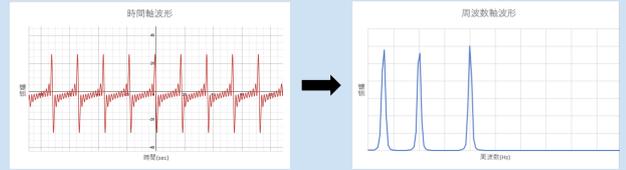
量子フーリエ変換における量子状態準備の方法の検討と実装

大月優佳 (広尾学園高等学校 2年)

概要

フーリエ変換とは、ある関数や信号にどのような周波数成分がどのくらいの大きさで含まれているかを把握する手法である。量子コンピュータ上でフーリエ変換を実行することで、従来の古典的なアルゴリズムではデータ量の増加に伴い計算量が指数関数的に増大する問題に対し、多項式時間で処理が可能となる。しかし、量子フーリエ変換を実現する量子回路は既に確立されている一方で、任意の数列に対して初期状態をどのように準備するかという問題は容易ではない。そこで、任意の数列を量子フーリエ変換の入力状態として設定する量子ビットの初期化方法を検討し、量子コンピュータ上で量子フーリエ変換を実行した。

フーリエ変換のイメージ

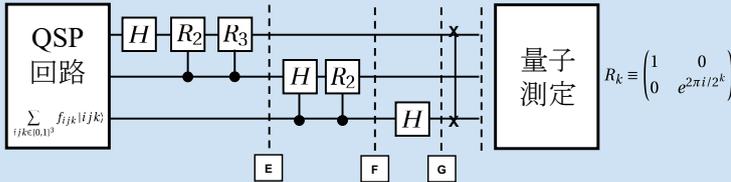


量子フーリエ変換

量子コンピュータを用いた高速で数列のスペクトル計算を実行するアルゴリズム

入力値 $|x\rangle := \sum_{j=0}^{2^n-1} x_j |j\rangle$ に対して出力値 $|y\rangle := \sum_{k=0}^{2^n-1} y_k |k\rangle$ の重み係数 y_k を
 複素数での離散フーリエ変換の式 $y_k = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} x_j e^{\frac{2\pi i k j}{2^n}}$ に変換する

3量子ビットにおける量子フーリエ変換の量子回路:

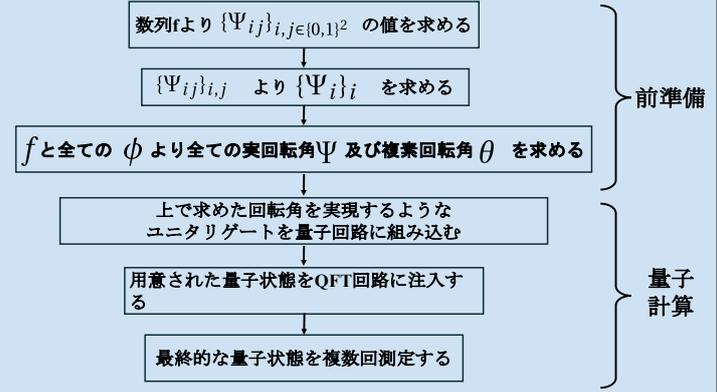


各々の時点での状態 $|ijk\rangle$, $i, j, k \in \{0, 1\}$ における量子ビットの状態

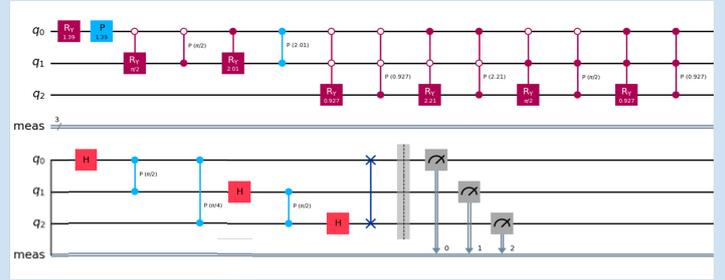
$$|ijk\rangle \rightarrow \begin{cases} \text{E} & (|0\rangle_3 + e^{2\pi i 0 \cdot ijk} |1\rangle_3) \otimes |j\rangle_2 \otimes |k\rangle_1 \\ \text{F} & (|0\rangle_3 + e^{2\pi i 0 \cdot ijk} |1\rangle_3) \otimes (|0\rangle_2 + e^{2\pi i 0 \cdot jk} |1\rangle_2) \otimes |k\rangle_1 \\ \text{G} & (|0\rangle_3 + e^{2\pi i 0 \cdot ijk} |1\rangle_3) \otimes (|0\rangle_2 + e^{2\pi i 0 \cdot jk} |1\rangle_2) \otimes (|0\rangle_1 + e^{2\pi i 0 \cdot k} |1\rangle_1) \end{cases}$$

量子フーリエ変換実行

量子フーリエ変換を実行するためのフローチャートと量子回路図:



Qiskitを用いて上記を組んだ量子回路:

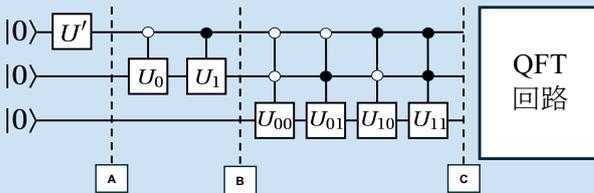


Quantum State Preparation

Grover-Rudolph Algorithm

任意の量子状態を制御回転ゲートを用いて用意するアルゴリズム

3量子ビットにおけるGrover-Rudolph Algorithmの量子回路:



各々の時点での量子ビットの状態

$$\begin{cases} \text{A} & \left(\sum_{i \in \{0,1\}} \Psi_i |i\rangle_1 \right) \otimes |0\rangle_2 \otimes |0\rangle_3 \\ \text{B} & \left(\sum_{(i,j) \in \{1,0\}^2} \Psi_{ij} |i\rangle_1 \otimes |j\rangle_2 \right) \otimes |0\rangle_3 \\ \text{C} & \sum_{(i,j,k) \in \{1,0\}^3} \Psi_{ijk} |i\rangle_1 \otimes |j\rangle_2 \otimes |k\rangle_3 \end{cases}$$

$$U_\alpha = R(\theta_\alpha) P(\phi_\alpha)$$

$$R_y(\theta) |0\rangle = \cos(\theta/2) |0\rangle + \sin(\theta/2) |1\rangle$$

$$P(\phi) |0\rangle = |0\rangle + e^{i\phi} |1\rangle$$

2変数が漸化的にどのように決定するかを表している (下図)

各々の量子ゲートにおける2変数と各々の量子ビットの重み係数

$$\Psi_{ij} = \arg(f_{ij0}) \sqrt{f_{ij0}^2 + f_{ij1}^2}$$

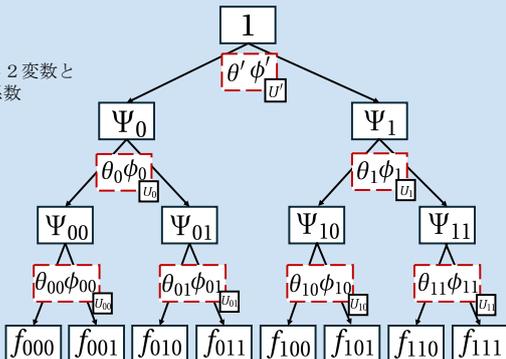
$$\Psi_i = \arg(\Psi_{i0}) \sqrt{\Psi_{i0}^2 + \Psi_{i1}^2}$$

$$\phi_i = \arg(\Psi_{i1}) - \arg(\Psi_{i0})$$

$$\phi_{ij} = \arg(f_{ij1}) - \arg(f_{ij0})$$

$$\theta_{ij} = \arctan \frac{|f_{ij1}|}{|f_{ij0}|}$$

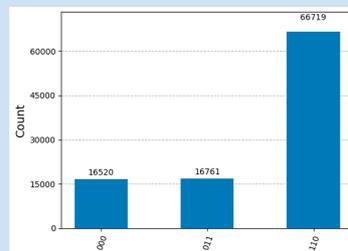
$$\theta_i = \arctan \frac{|\Psi_{i1}|}{|\Psi_{i0}|}$$



結果

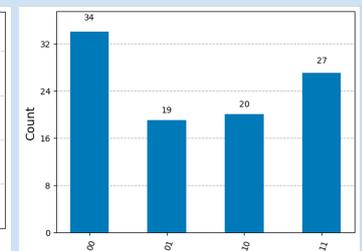
<シミュレーションでのQSPの結果>

数値列 $f = [1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0]$ を3量子ビットを用いて表現できた



<量子コンピュータで実行したQFTの結果>

数値列 $f = [1, 0, 0, 0, 0, 0, 0]$ に対して2量子ビットを用いて量子フーリエ変換を実行した



展望

量子ビット数を増やし、より長い数列に対しての量子フーリエ変換をシミュレーションおよび実際の量子コンピュータ上で実行する。より大規模な任意の量子状態にも対応できるような量子状態準備の方法として、テンソルネットワークを用いたQSPを取得し、最終的にQSVTなどを含む最先端の量子アルゴリズムへの理解を深める。

参考文献

- 石坂 智, 小川 朋宏, 河内 亮周, 木村 元, 林 正人, 2012. 量子情報入門, 共立出版
- Debra Ramacciotti, Andreea I. Lefterovic, and Antonio F. Rotundo(2024), Simple quantum algorithm to efficiently prepare sparse states, PHYSICAL REVIEW A 110, 032609
- Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press
- Huang, H.-Y., Kueng, R., & Preskill, J. (2023). Efficient learning of quantum dynamics with a neural network. Quantum Science and Technology, 8(4), 045017.