

理論演習 II Essay

05-201564 Shun Yamasaki

2022 年 1 月 13 日

1 Abstract

線形連立一次方程式を解くことは周知の通り様々な場面で非常に重要な問題である [1]。線形連立一次方程式は

$$A\vec{x} = \vec{b} \tag{1}$$

と表される。今回は、 \vec{x} 自体を知る必要はないが、ある行列 M を用いて $\vec{x}^\dagger M \vec{x}$ と表されるような \vec{x} に関係した量を求めたい場合を考える。 A が行あたり最大 s 個の non-zero 成分を持つ $N \times N$ の行列であり、条件数 (condition number)^{*1} を κ とすると、知られている最速の古典アルゴリズムでは $\vec{x}, \vec{x}^\dagger M \vec{x}$ を計算するのに $N\sqrt{\kappa}$ の時間スケールで計算できる。一方、今回紹介する量子アルゴリズムでは \vec{b} に対応する状態 $|b\rangle$ が用意できた時、 $|x\rangle$ を求めるまでの計算ステップは $\log(N), \kappa$ の多項式時間スケールで計算することができる。これは古典コンピュータに比べて指数関数的なスピードアップである。

2 Introduction

量子コンピュータは量子力学を利用し、現状わかっている古典コンピュータによるアルゴリズムでは実現できない計算を含む計算を実現する装置である。Shor のアルゴリズムのように、ある特定の問題に対しては量子コンピュータは古典コンピュータよりはるかに早い時間で計算できる。この論文では線形連立一次方程式の解の特徴について考えていく。

線形方程式は科学技術・工学においてほぼ全ての分野で応用 [1] され、非常に重要であると言える。古典コンピュータでは N 次の線形方程式を解くのに、一般的に少なくとも N オーダーの時間がかかる。

^{*1} condition number は線形方程式の解の不確かさの上限を表し、 x の相対誤差を b の相対誤差で割った最大比率で定義される。

この論文では、特定の場合では Abstract と同じ方程式

$$A\vec{x} = \vec{b} \quad (2)$$

において A が行あたり最大 s 個の non-zero 成分を持つ $N \times N$ の行列であり、条件数 (condition number) を κ とすると、解 \vec{x} に関連した値を任意の精度で計算するのに $\log(N), \kappa$ の多項式時間スケールかかるということを示している。これは古典コンピュータに比べて指数関数的なスピードアップである。また、典型的なケースでは正確性はあまり求められないことが多い。しかし、condition number は計算量を著しく増大させ得る。これはこの論文におけるアルゴリズムにおける強い制約となっている。

上の線形方程式と同じ条件で以下、考えていくこととする。まず、 \vec{b} は抽象的な状態 $|b\rangle$ の行列表現であると考える。 \vec{b} と $|b\rangle$ が必ずしも等価でないことに注意する。

また、related work として制限付きでの線形操作の例や、非線形微分方程式への拡張の話がある。

3 Preliminary

3.1 量子計算の基礎知識

量子ビットは量子情報の入れ物の最小単位で、2 準位基底では $|0\rangle, |1\rangle$ を基底に取り一般の状態 $|\psi\rangle$ を $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ のように表すことが多い。ただし、 α, β は複素数の定数で、 $|\alpha|^2 + |\beta|^2 = 1$ である。

量子レジスタは異なる Hilbert space に用意された量子ビットを複数並べたもので、ここではテンソル積を用いて $|0\rangle \otimes |1\rangle \otimes \cdots \otimes |1\rangle$ のように表すものとする。

量子ゲートは量子レジスタに対する操作のことで、現在の主流な量子コンピュータでは数種類の unitary operator の組み合わせ (これも unitary operator になることが知られている) によって望みの状態を実現する。

量子回路は、量子レジスタにゲートを順番にかけていった回路図で、以下の図のように表される。

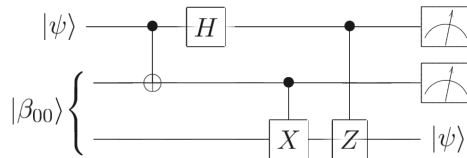


図1 quantum circuit の例 (QCQI より引用)

measurement は終状態から情報を読み出す操作である。終状態 $\sum_{k=0}^{2^n-1} c_k |k\rangle$ から $p_k = |c_k|^2$ を知ることを指し、回路の実行・測定を繰り返し、 $0, 1, \dots, 2^n-1$ を得た回数を記録しヒストグラムから $\{p_k\}$ を推定する。

4 Main Result

具体的なアルゴリズムを紹介する。

1. [2] によって行列 A を e^{iAt} に変換する。
2. [3] によって \vec{b} に対応する状態 $|b\rangle$ を用意する。
3. 同様に [3] によって $|\Psi_0\rangle := \sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin \frac{\pi(\tau + \frac{1}{2})}{T} |\tau\rangle$ なる状態 $|\Psi_0\rangle$ を用意する。
4. $\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau| \otimes e^{iA\tau t_0/T}$ (unitary) on $|\Psi_0\rangle \otimes |b\rangle$, where $t_0 = O(\kappa/\epsilon)^{*2}$ のように用意した状態 $|b\rangle$ および $|\Psi_0\rangle$ のテンソル積に対して conditional Hamiltonian evolution を作用させる。
5. first register(initial state が $|\Psi_0\rangle$ の register) を Fourier Transformation する。
6. qubit を追加して $\tilde{\lambda}_k$ の値に依存して rotation させる。
7. 逆 Fourier Transformation する。
8. 4 の逆変換をする。
9. last qubit を measure する。1 が出れば成功で 0 が出れば失敗である。

以上のアルゴリズムによって、 \vec{x} に関係した量を \vec{b} に対応する状態 $|b\rangle$ が用意できた時、残りの $\vec{x}^\dagger M \vec{x}$ を $\log(N), \kappa$ を求める計算ステップは多項式時間スケールで計算することができる。

*2 epsilon の説明を入れる

5 Details, Discussion

5.1 QFT

QFT(Quantum Fourier transformation) は以下のように状態 $|j\rangle$ を変換する操作である。

$$\begin{aligned}
 |j\rangle &\rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle \\
 &= \frac{1}{2^{n/2}} \sum_{k_1=0}^1 \cdots \sum_{k_n=0}^1 e^{2\pi i j (\sum_{l=1}^n k_l 2^{-l})} |k_1 \dots k_n\rangle \\
 &= \frac{1}{2^{n/2}} \sum_{k_1=0}^1 \cdots \sum_{k_n=0}^1 \bigotimes_{l=1}^n e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
 &= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n \left[\sum_{k_l=0}^1 e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \\
 &= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right] \\
 &= \frac{(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle) (|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle) \cdots (|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \cdots j_n} |1\rangle)}{2^{n/2}}.
 \end{aligned}$$

この計算は具体的に、以下のような回路で実現することができる。

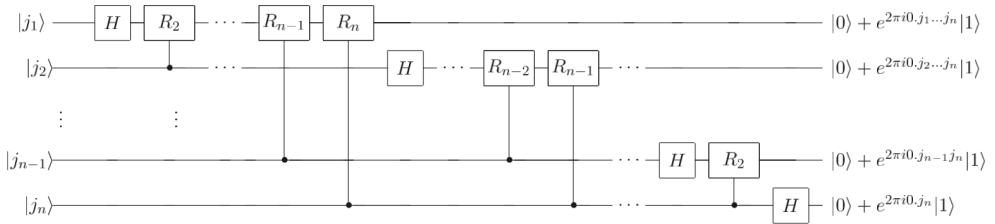


図2 QFT の例 (QCQI より引用)

5.2 QPE

QPE(Quantum Phase Estimation) は位相を推定するためのアルゴリズムで、QFT を応用して計算する。具体的なアルゴリズムは以下の通りである。(QCQI より引用)

Inputs:

- (1) A black box which performs a controlled- U^j operation, for integer j
- (2) an eigenstate $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi_u}$
- (3) $t = n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$ qubits initialized to $|0\rangle$.

Outputs: An n -bit approximation $\widetilde{\varphi}_u$ to φ_u .

Runtime: $O(t^2)$ operations and one call to controlled- U^j black box. Succeeds with probability at least $1 - \epsilon$.

Procedure:

1. $|0\rangle|u\rangle$
2. $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|u\rangle$ create superposition
3. $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i j \varphi_u} |j\rangle|u\rangle$ apply black box
4. $\rightarrow |\widetilde{\varphi}_u\rangle|u\rangle$ apply inverse Fourier transform
5. $\rightarrow \widetilde{\varphi}_u$ measure first register

これを量子回路で実現すると以下ようになる。

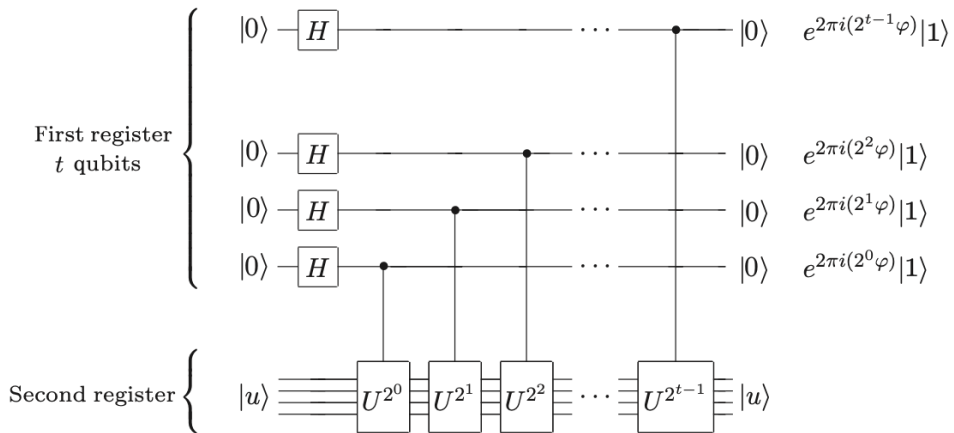


図3 QPE の一部の例 (QCQI より引用)

5.3 アルゴリズムの解説

以下では Main Result で紹介したアルゴリズムがうまくいく理由について解説する。まず、[2] で、行列 A を $O(\log(N)s^2t)$ を upper bound とする計算量で $\exp(iAt)$ で計算することができる。

次に、行列 A が Hermitian かどうかで場合分けをする。

A が Hermitian でない場合、

$$\tilde{A} := \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \quad (3)$$

と定義すれば \tilde{A} は Hermitian であるので、方程式

$$\tilde{A}\vec{y} = \begin{pmatrix} \vec{b} \\ 0 \end{pmatrix}$$

を解くことで

$$\vec{y} = \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix}$$

が得られる。このように、 A が Hermitian でない時、 \tilde{A} を定義して解けるようにする一連の操作を reduction と呼ぶ。

以下では、 A が Hermitian である場合を考える。[3] により、 b_i and $\sum_{i=i_1}^{i_2} |b_i|^2$ are efficiently computable なら state $|b\rangle$ を任意の基底における任意の確率振幅についての重ね合わせに変換できる。ここでは $|b\rangle$ を A の固有ベクトル (単位ベクトルを選ぶ) をの重ね合わせで表すこととする。確率振幅には、 \vec{b} の成分を割り当てることを考える。これでまた、 $|u_j\rangle$ を the eigenvectors of A , and λ_j を the corresponding eigenvalues とし、次を定義する。ここまですべて $|b\rangle$ が準備できた。

$$|\Psi_0\rangle := \sqrt{\frac{2}{T}} \sum_{\tau=0}^{T-1} \sin \frac{\pi(\tau + \frac{1}{2})}{T} |\tau\rangle$$

ただし、ここでは T は非常に大きな値とする。これで $|\Psi_0\rangle$ についても準備ができた。このようにして用意した $|b\rangle$ および $|\Psi_0\rangle$ のテンソル積に対して conditional Hamiltonian evolution を作用させて

$\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau| \otimes e^{iA\tau t_0/T}$ (unitary) on $|\Psi_0\rangle \otimes |b\rangle$, where $t_0 = O(\kappa/\epsilon)^{*3}$ を考える。ただし、

*3 epsilon の説明を入れる

$t_0 = O(\kappa/\epsilon)$ とする。first register(initial state が $|\Psi_0\rangle$ の register) を Fourier Transformation すると, first register での state は

$$\sum_{j=1}^N \sum_{k=0}^{T-1} \alpha_{k|j} \beta_j |k\rangle |u_j\rangle$$

となる。 $|\tilde{\lambda}_k\rangle$ を用いて $|k\rangle$ を置き換えて

$$\sum_{j=1}^N \sum_{k=0}^{T-1} \alpha_{k|j} \beta_j |\tilde{\lambda}_k\rangle |u_j\rangle$$

とできる。qubit を追加して $|\tilde{\lambda}_k\rangle$ に回転を施して,

$$\sum_{j=1}^N \sum_{k=0}^{T-1} \alpha_{k|j} \beta_j |\tilde{\lambda}_k\rangle |u_j\rangle \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_k^2}} |0\rangle + \frac{C}{\tilde{\lambda}_k} |1\rangle \right)$$

ただし C は $O(1/\kappa)$ 。phase estimation で $|\tilde{\lambda}_k\rangle$ を逆算できる。phase estimation が perfect なら,

$$\alpha_{k|j} = 1 \text{ if } \tilde{\lambda}_k = \lambda_j, \text{ and } 0 \text{ otherwise}$$

すると,

$$\sum_{j=1}^N \beta_j |u_j\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right)$$

last qubit を measure すれば,

$$\sqrt{\frac{1}{\sum_{j=1}^N C^2 |\beta_j|^2 / |\lambda_j|^2}} \sum_{j=1}^N \beta_j \frac{C}{\lambda_j} |u_j\rangle$$

が得られ, これは, $|x\rangle = \sum_{j=1}^n \beta_j \lambda_j^{-1} |u_j\rangle$ に対応。 M で measure すれば $\langle x|M|x\rangle$ が得られる。

6 Conclusion

線形連立方程式 $A\vec{x} = \vec{b}$ を解くことはあらゆる分野で重要な問題であり, \vec{x} 自体を知る必要はないが, ある行列 M を用いて $\vec{x}^\dagger M \vec{x}$ と表されるような \vec{x} に関係した量を求めたい場合, 量子位相推定を応用することで確率的に求めることができる。また, その成功確率は condition number κ に依存する。 \vec{b} に対応する状態 $|b\rangle$ が用意できた時, $|x\rangle$ を求める計算ステップは $\log(N), \kappa$ の多項式時間スケールで計算することができる。これは古典的なコンピュータでのわかっている範囲で最も速いアルゴリズムと比較して指数関数的な速度の向上である。

7 Reference

参考文献

- [1] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. Physical Review Letters, 103(15):1–8, 2009.
- [2] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. Communications in Mathematical Physics, 270(2):359–371, 2007.
- [3] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. pages 1–2, 2002.