日本語訳『Qiskit Textbook』勉強会 第4章 4.1.1

Takehiko Amano
IBM Garage / Tokyo lab.

HHL(Harrow-Hassidim-Lloyd)アルゴリズムについて

- 線型方程式系 (連立一次方程式)の解を高速に算出するアルゴリズム。
  - 線型方程式は、偏微分方程式、電磁気・流体シミュレーション、金融モデルに現れるが、高速に実行できるため着目を浴びています。
  - 古典計算(共役勾配法など) が $\mathcal{O}(Ns\kappa\log(1/\epsilon))$ の時間に対して、$\mathcal{O}(\log(N)s^2\kappa^2/\epsilon)$ で計算ができます。

注意点：

- 行列は疎(スパース)でないといけない
- 行列はエルミートである必要がある（エルミートにできる）
- 厳密解ではなく近似解として解が出てくる。

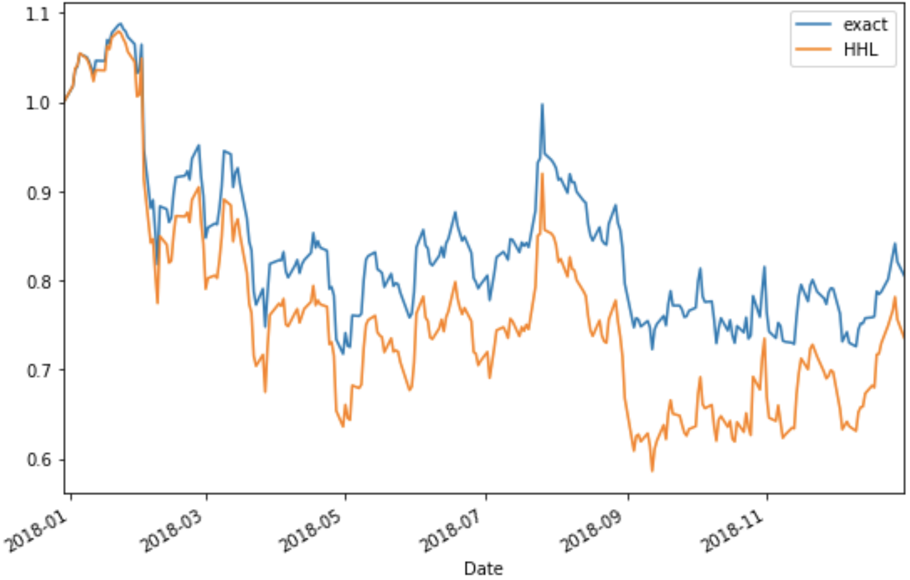
エルミートでない場合は以下のようにする

$$\tilde{A} = \begin{pmatrix} O & A \\ A^\dagger & O \end{pmatrix}, \tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}$$

https://dojo.qulacs.org/ja/latest/notebooks/7.2_Harrow-Hassidim-Lloyd_algorithm.html から抜粋

HHLの応用 – ポートフォリオ分析

過去の株価変動のデータから、最適なポートフォリオ（資産配分）ができるようです。



左図はポートフォリオの価格変動予測

https://dojo.qulacs.org/ja/latest/notebooks/7.3_application_of_HHL_algorithm.html から抜粋

連立一次方程式例

- Ax = b の時 x を求める。x = A^{-1}b
- 教科書例：

$$A = \begin{pmatrix} 1 & 1/3 \\ 1/3 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

答え

$$x = A^{-1}b = \frac{3}{8}\begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 9/8 \\ 3/8 \end{pmatrix}$$

$$\text{固有値}：\lambda_1 = \frac{4}{3}\ \lambda_2 = \frac{2}{3}, \text{固有ベクトル}\begin{pmatrix} -1 \\ 1 \end{pmatrix}\ \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

答えを固有ベクトル
で展開してみます

$$-\frac{1}{2\lambda_1}\begin{pmatrix} -1 \\ 1 \end{pmatrix} + \frac{1}{2\lambda_2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{3}{8}\begin{pmatrix} 1 \\ -1 \end{pmatrix} + \frac{3}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 9/8 \\ 3/8 \end{pmatrix}$$

この係数がHHLのキモ

基本的な考え方(1/1)

1. x b を量子状態にマッピングする

$$A|x\rangle = |b\rangle$$

2. A を 固有値と固有ベクトルでスペクトル分解する

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle\langle u_j|, \quad \lambda_j \in \mathbb{R}$$

3. |b> をAの固有ベクトルで展開する

$$|b\rangle = \sum_{j=0}^{N-1} b_j |u_j\rangle, \quad b_j \in \mathbb{C}$$

|b> |x> などは正規化できると仮定する。

基本的な考え方(2/2)

4. A の逆行列を求める

$$A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1} |u_j\rangle\langle u_j|$$

5. |x> を求める

$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} \boxed{\frac{b_j}{\lambda_j}} |u_j\rangle$$

これが解になっている！

一般的に以下の方程式が成立する

λが固有値、ψが固有ベクトルの場合

$$f(A)|\psi\rangle = f(\lambda)|\psi\rangle$$

問題は A の固有値と固有ベクトルを求める問題になる（実際のところは固有値だけで事足りる）

Aの固有値？

量子位相推定(Quantum Phase Estimation – QPE)

- **量子位相推定(Quantum Phase Estimation)** (省略して **QPE**) アルゴリズムは、U に対応するユニタリーゲートと状態 $|0\rangle_n|\psi\rangle_m$を入力として、状態 $|\tilde{\theta}\rangle_n|\psi\rangle_\mathrm{m}$を返すアルゴリズムでした(Qiskit textbook 第３章 3.8)

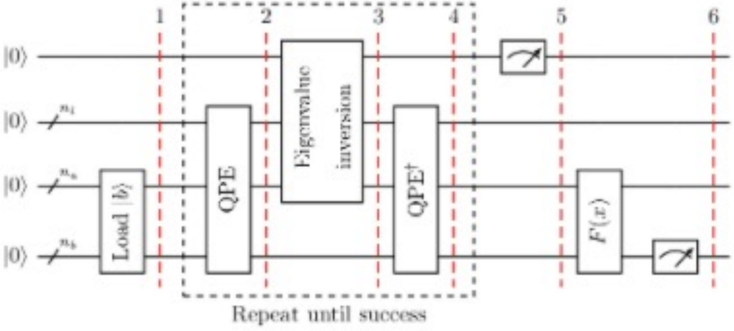
- HHL ではユニタリーとして $U=e^{iAt}$ を利用します(Aは全ページの行列)。

- 仮にそれぞれの λ_j が n_l ビットで正確に記述できる場合には

$$\mathrm{QPE}\left(e^{iA2\pi}, \sum_{j=0}^{N-1} b_j|0\rangle_{n_l}|u_j\rangle_{n_b}\right) = \sum_{j=0}^{N-1} b_j|\lambda_j\rangle_{n_l}|u_j\rangle_{n_b}.$$

固有値が出てきました！！

HHL アルゴリズムの概略(1/2)



1. $|b\rangle$をロードする(次の変換を実施する）

$$|0\rangle_{n_b} \mapsto |b\rangle_{n_b}$$

2. QPE を適用する。全体の状態は次のようになる。

$$\sum_{j=0}^{N-1} b_j |\lambda_j\rangle_{n_l} |u_j\rangle_{n_b}$$

3. 補助ビットを使って、制御回転を実施。

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

λ_j を補助ビットに押し出すイメージ

C は規格化定数

HHL アルゴリズムの概略(2/2)

4. QPE† を適用します。QPE で発生しうるエラーを無視すると、次の結果になります。

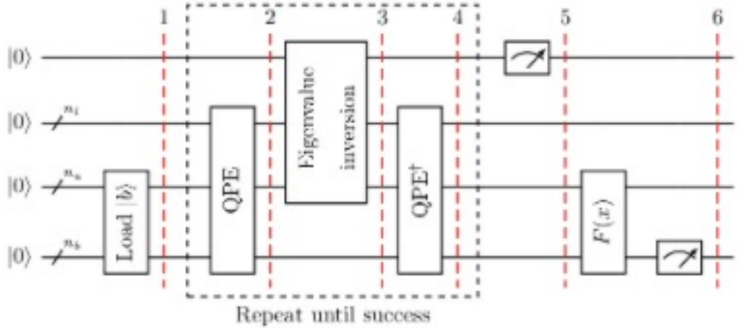
$$\sum_{j=0}^{N-1} b_j |0\rangle_{n_l} |u_j\rangle_{n_b} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right)$$

ここがゼロに戻る

5. 計算基底にて補助量子ビットを測定します。結果が 1 の場合、測定後のレジスターの状態は次のようになります。

$$\left(\sqrt{\frac{1}{\sum_{j=0}^{N-1} |b_j|^2/|\lambda_j|^2}} \right) \sum_{j=0}^{N-1} \boxed{\frac{b_j}{\lambda_j}} |0\rangle_{n_l} |u_j\rangle_{n_b},$$

規格化定数を除き、解になってます！



$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} \boxed{\frac{b_j}{\lambda_j}} |u_j\rangle$$

冒頭の例

$$A = \begin{pmatrix} 1 & 1/3 \\ 1/3 & 1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

演算は省略しますが、結果は

$$\frac{\frac{3}{2\sqrt{2}}|u_1\rangle + \frac{3}{4\sqrt{2}}|u_2\rangle}{\sqrt{45/32}} = \frac{|x\rangle}{||x||}$$

規格化定数を除き答えが出ています

$$-\frac{1}{2\lambda_1}\begin{pmatrix} -1 \\ 1 \end{pmatrix} + \frac{1}{2\lambda_2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{3}{8}\begin{pmatrix} 1 \\ -1 \end{pmatrix} + \frac{3}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 9/8 \\ 3/8 \end{pmatrix}$$

Qiskit での実装

- Qiskit Aqua に HLL アルゴリズムがライブラリとして用意されているのでそれを利用します。

基本的には行列 A と ベクトル |b> があればよい

```
class HHL(matrix, vector, truncate_powerdim=False,
truncate_hermitian=False, eigs=None, init_state=None, reciprocal=
None, num_q=0, num_a=0, orig_size=None,
quantum_instance=None)
```

$$\begin{pmatrix} 0 & A^{\mathrm{H}} \\ A & 0 \end{pmatrix}$$

truncate_hermitian は A がハミルトニアンでない場合に利用する

Qiskit エミュレーター実行結果

```
Solution:                    [1.13586-0.j 0.40896+0.j]
Classical Solution:          [1.125 0.375]
Probability:                 0.056291
Fidelity:                    0.999432
```

フィデリティーは qiskit.quantum_info.state_fidelity メソッドを利用して計算

なお 時間として $t = 2\pi \cdot \frac{3}{8}$, を入れるとフィデリティーは 1 になる

（横着して固有値を計算してあらかじめ求めておいて、t の値を逆計算するイメージ）

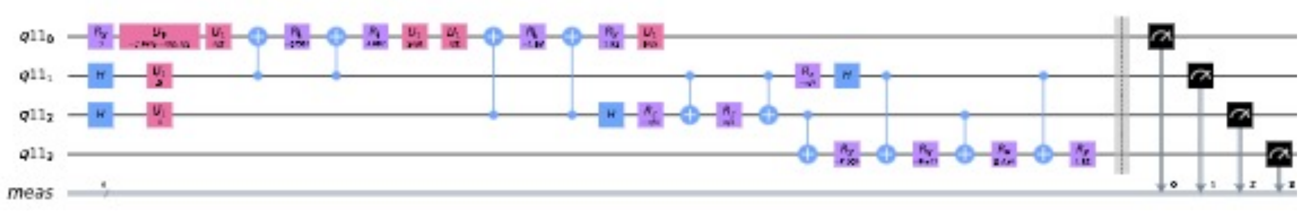
実デバイスで動作させる場合の課題

- 一般的に量子位相推定の回路は深さ (depth) が大きく、CNOT も多用するためノイズのあるデバイス(NISQ) では近似が難しいと言われています。
- そこで実デバイスでも深さやCNOT の数を減らし NISC でも動作するアルゴリズムの研究がなされているようです。
  - Richardson extrapolation (リチャードソンの補外) による “Quantum Linear System Algorithm” の拡張 (A. Carrera Vazquez, A. Frisch, D. Steenken, H. S. Barowski, R. Hiptmair, and S. Woerner)
  - UniversalQCompiler による回路最適化

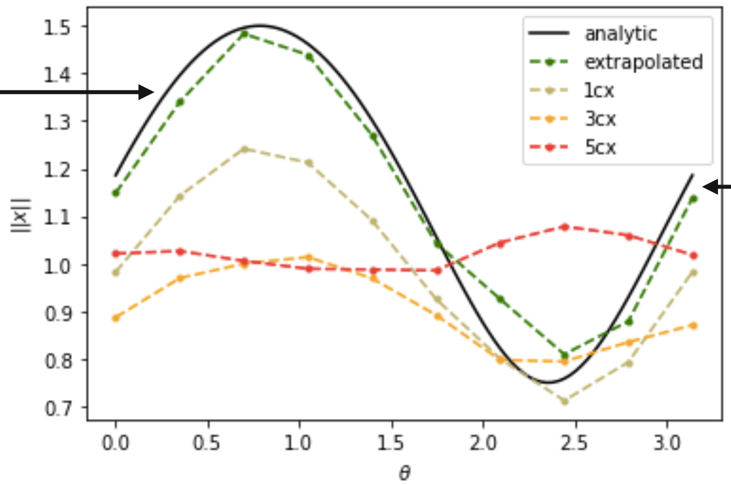
実デバイスでの HHL

- ノイズのリチャードソン外挿や最適化を施すと回路の深さやCNOT数が軽減します。

```
Depth: 26
CNOTS: 10
```



厳密解



外挿した結果の近似解

その他

- 変分法を利用した線型方程式の計算ができました。

  Variational Quantum Linear Solver

- てなわけで、4.2.1 章に続きます・・ To be continued !

参考文献

1. Harrow-Hassidim-Lloyd (HHL) アルゴリズム
   https://dojo.qulacs.org/ja/latest/notebooks/7.2_Harrow-Hassidim-Lloyd_algorithm.html

2. 連立一次方程式を量子コンピューターで解く - HHLアルゴリズム
   https://whyitsso.net/physics/quantum_mechanics/HHL.html

Takehiko Amano