Qiskit Textbook 4.2.1
Variational Quantum Linear Solver 解説

Ayumu Shiraishi

Qiskit Advocate

Quantum Tokyo

Agenda

- VQLSとは
  - 連立一次方程式
  - 変分法のおさらい
  - 特徴
- VQLSの流れ
  - 準備
  - コスト関数
  - 変分パラメーター
- デモ
- 読んでいて出てきた疑問
- まとめ

VQLSとは

連立一次方程式（Linear System）を解くために量子回路（Quantum Circuit）に変分法（Variational method）を適用した手法（Solver）

Variational **Q**uantum **L**inear **S**olver

VQEのアイデアを元にした手法

古典よりも効率的に解ける（らしい）

連立一次方程式とは

例：

$$\begin{cases} 3x + y = 1 \\ 5x + 2y = 1 \end{cases}$$

$$x = 1, y = -2$$

n個の求めたい変数$x_k (k = 1,2, \dots, n)$に対して、同時に成立するn個の方程式がある場合

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n = b_2 \\ \quad\quad\quad\quad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots a_{nn}x_n = b_n \end{cases}$$

ベクトルと行列で表現すると

$$\boxed{Ax = b}$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

参考：連立一次方程式の補足

変数xを求めるための様々な解法が知られている

（・鶴亀算）

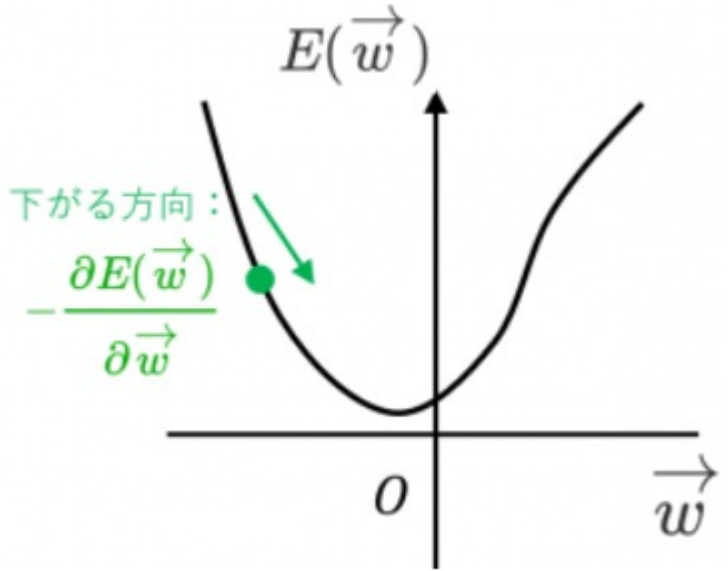
・Gaussの消去法：計算量O(n^3)

・LU分解：計算量O(n^2)

連立一次方程式は、物理現象や社会現象を立式すると出てくることが多い

変分法のおさらい

損失関数、コスト関数などを定義し、変分パラメーターを調整しながら、最小値の損失（ないしはコスト）を探していくアプローチ



VQLSの特徴

- HHLアルゴリズムと同様に連立一次方程式を解くアルゴリズム
  - 古典コンピューターよりも効率的に解けるとされている
  - VQLSはNISQデバイスで動作することが可能であるのに対し、HHLアルゴリズムは十分なエラー耐性のあるデバイスで動作することが前提になっている
    - ただし、HHLの方が性能では上回るとされている

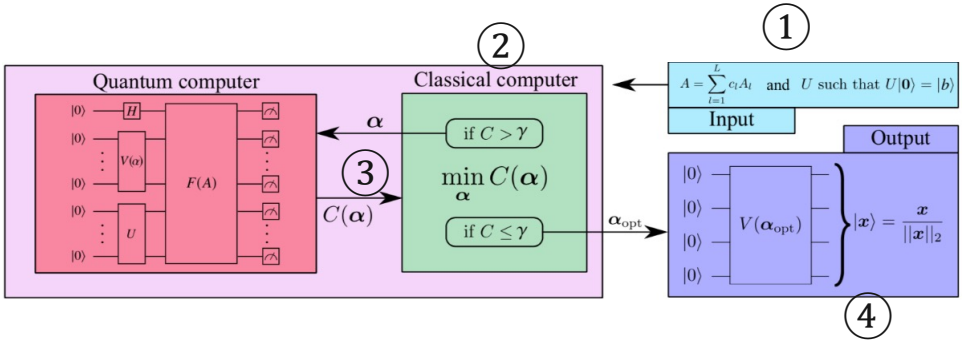
VQLSの流れ

① Aをユニタリ行列の線形和に分解し、$|b\rangle = U|0\rangle$となるUおよびコスト関数も定義しておく

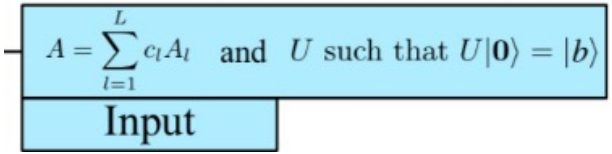
② 任意に選んだ初期変分パラメーターαを使って量子状態$|x\rangle$を生成しHadamardテストを実施した結果から、古典コンピューターでコスト関数を計算する

③ コスト関数が閾値γよりも大きい場合は、αの値を変分法により再計算し、新たに量子回路に入れて計算を行い、これを閾値よりも小さくなるまで繰り返す

④ コスト関数の値が、閾値以下になったαをα_{opt}として、再度状態$|x\rangle$を作る回路に投入することで、求めたい連立一次方程式の解xの規格化された量子状態$|x\rangle$を得る



①準備



連立一次方程式のインプットになる既知の行列Aおよび、ベクトルbについて量子コンピューターで扱えるように準備を行う。

- Aをユニタリ行列の線形和に分解する。すなわち、

$$A = \sum_n c_n A_n \quad （A_n\text{はユニタリ行列、}c_n\text{は複素係数}）$$

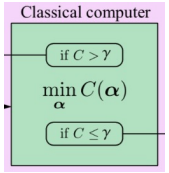
- $|0\rangle$から$|b\rangle$に遷移するためのユニタリ行列をUとする。すなわち、

$$|b\rangle = U|0\rangle$$

- $|x\rangle$の規格化したベクトルを$|\psi\rangle$とし、 $|0\rangle$ から$|\psi\rangle$ に遷移するユニタリ行列をVとする。すなわち、

$$|\psi\rangle = \frac{|x\rangle}{|x|} = V|0\rangle$$

②コスト関数Cの定義と導出方法



コストは**「$|\Phi(k)\rangle = A|\psi(k)\rangle$がどれくらい$|b\rangle$に近いか」**という思想で算出する。

そのために、射影演算子を以下のように定義する。

$$H_p = I - |b\rangle\langle b|$$

状態$|\Phi(k)\rangle$での平均値によってコストを決めることにする。つまり、

$$C_P = \langle\Phi(k)|H_p|\Phi(k)\rangle = \langle\Phi(k)|\Phi(k)\rangle - \langle\Phi(k)|b\rangle\langle b|\Phi(k)\rangle$$

ただし、$|\Phi(k)\rangle$のノルム（長さ）が小さいと、全体としてコストも小さく見積もられてしまうため、それを避けるために全体を$\langle\Phi(k)|\Phi(k)\rangle$で割る。つまり、

$$\widehat{C_P} = \frac{C_P}{\langle\Phi(k)|\Phi(k)\rangle} = 1 - \frac{\langle\Phi(k)|b\rangle\langle b|\Phi(k)\rangle}{\langle\Phi(k)|\Phi(k)\rangle} = 1 - \frac{|\langle b|\Phi(k)\rangle|^2}{\langle\Phi(k)|\Phi(k)\rangle}$$

次からは$\widehat{C_P}$の２項目の分母と分子をそれぞれ分けて求め方を解説していく。

分子部分 $\langle\Phi(\mathrm{k})|\Phi(\mathrm{k})\rangle$の算出方法

まずは、分子の$\langle\Phi(\mathrm{k})|\Phi(\mathrm{k})\rangle$を$|\Phi(k)\rangle = A|\psi(k)\rangle = AV(k)|0\rangle$と$A = \sum_n c_n A_n$ を使って展開すると

$$\langle\Phi(\mathrm{k})|\Phi(\mathrm{k})\rangle = \langle\psi(k)|\, A^\dagger A\, |\psi(k)\rangle = \langle 0|V(k)^\dagger A^\dagger A\, V(k)|0\rangle$$

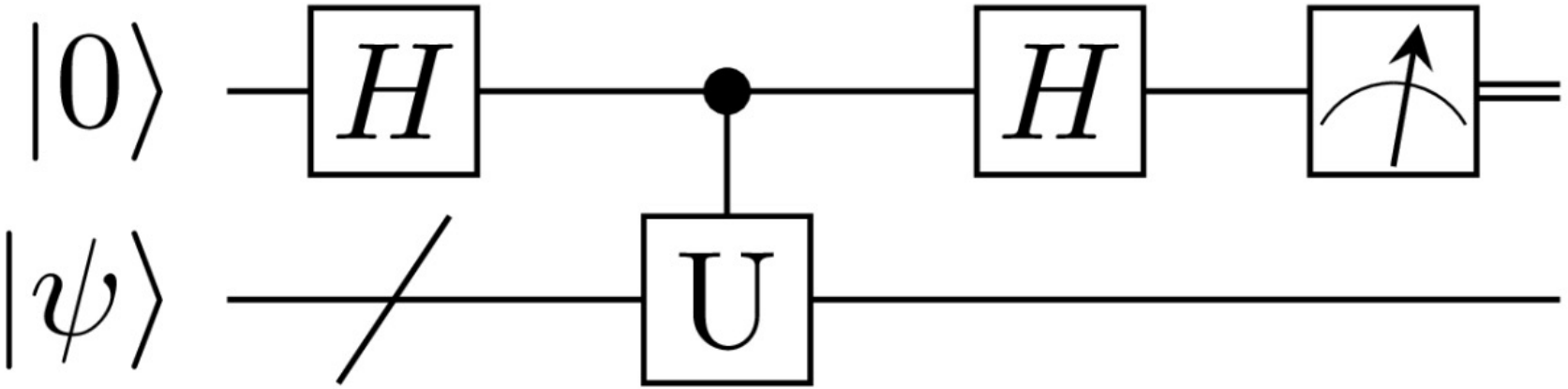
$$= \sum_m \sum_n c_m^* c_n \langle 0|V(k)^\dagger\, A_m^\dagger\, A_n V(k)|0\rangle = \sum_m \sum_n c_m^* c_n \langle\psi(k)|\, A_m^\dagger\, A_n|\psi(k)\rangle$$

ユニタリ行列

「量子状態$|0\rangle$に対するユニタリ行列$V(k)^\dagger\, A_m^\dagger A_n\, V(k)$の期待値」
または「量子状態$|\psi(k)\rangle$に対するユニタリ行列$A_m^\dagger A_n$の期待値」とみなせて、
それらに複素係数をかけた和から求めることができる。

Hadamardテスト

「状態$|\psi\rangle$に対するユニタリ行列Uの期待値」を測るための実装方法



Hadamardテスト

Hadamardテストでの期待値（の実部）の求め方は以下の通り

$|0\rangle \otimes |\psi\rangle \xrightarrow{H \otimes I} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle) \xrightarrow{\text{Controlled-U}} \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U|\psi\rangle)$

$\xrightarrow{H \otimes I} \frac{1}{2}(|0\rangle + |1\rangle) \otimes |\psi\rangle + \frac{1}{2}(|0\rangle - |1\rangle) \otimes U|\psi\rangle = \boxed{|0\rangle \otimes \frac{(I+U)}{2}|\psi\rangle + |1\rangle \otimes \frac{(I-U)}{2}|\psi\rangle}$

第1量子ビットに“0”が観測される確率

$$P(0) = \frac{1}{4}\langle\psi|(I+U^\dagger)(I+U)|\psi\rangle = \frac{1}{4}\langle\psi|(I^2 + U^\dagger + U + U^\dagger U)|\psi\rangle = \frac{1}{4}\langle\psi|(2I + U^\dagger + U)|\psi\rangle$$

$$= \frac{1}{4}\left(2\langle\psi|\psi\rangle + \langle\psi|U^\dagger|\psi\rangle + \langle\psi|U|\psi\rangle\right) = \frac{1}{4}(2 + 2Re(\langle\psi|U|\psi\rangle)) = \frac{1}{2}(1 + Re(\langle\psi|U|\psi\rangle))$$

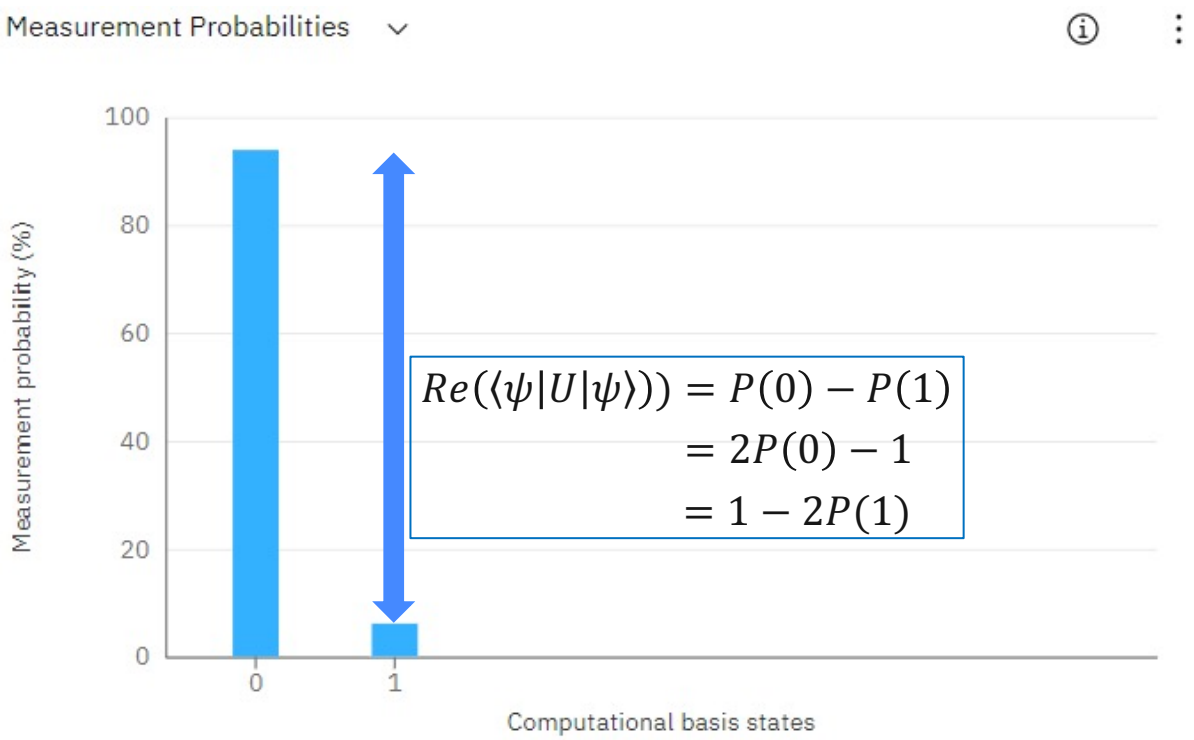
第1量子ビットに“1”が観測される確率は同様にして

$$P(1) = \frac{1}{2}(1 - Re(\langle\psi|U|\psi\rangle))$$

$$\boxed{Re(\langle\psi|U|\psi\rangle) = P(0) - P(1)}$$

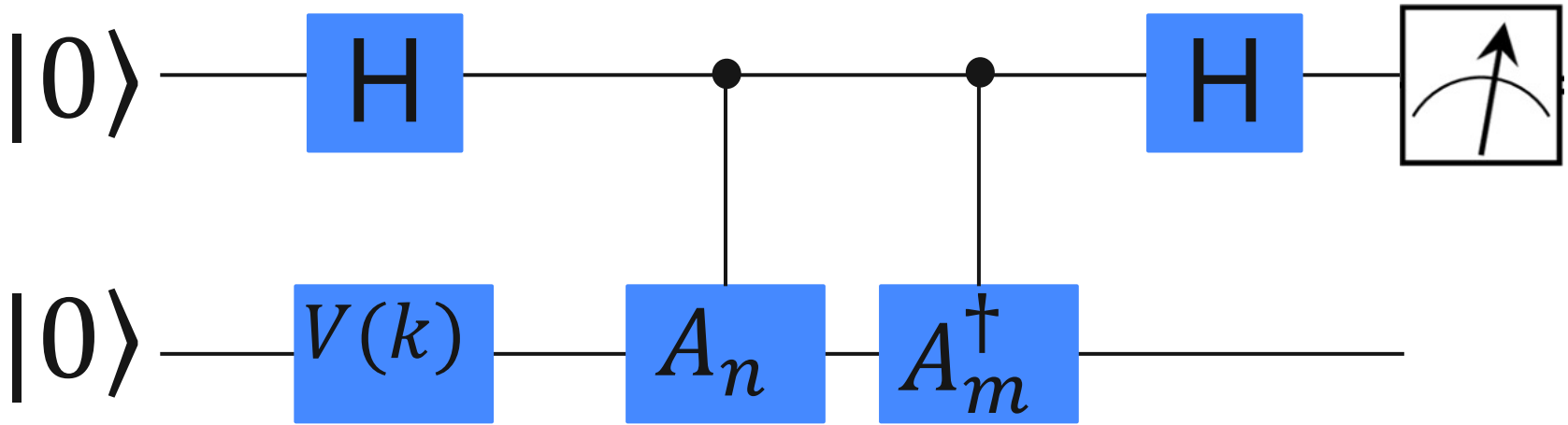
Hadamardテスト

HadamardテストでUの期待値（の実部）を視覚的に表現すると



$\langle 0|V(k)^\dagger A_m^\dagger A_n V(k)|0\rangle$の値を得る量子回路

Hadamardテストの回路を参考にするとこのようにすれば値が求まる



分母部分$|\langle b|\Phi(\mathrm{k})\rangle|^2$の導出方法

分子のときと同じように、分解していく。

$$|\langle b|\Phi(\mathrm{k})\rangle|^2 = |\langle b|AV(k)|0\rangle|^2 = |\langle 0|U^\dagger AV(k)|0\rangle|^2 = \langle 0|U^\dagger AV(k)|0\rangle\langle 0|V^\dagger A^\dagger U|0\rangle$$

$$= \sum_m \sum_n c_m^* c_n \langle 0|U^\dagger A_n V(k)|0\rangle\langle 0|V^\dagger A_m^\dagger U|0\rangle$$

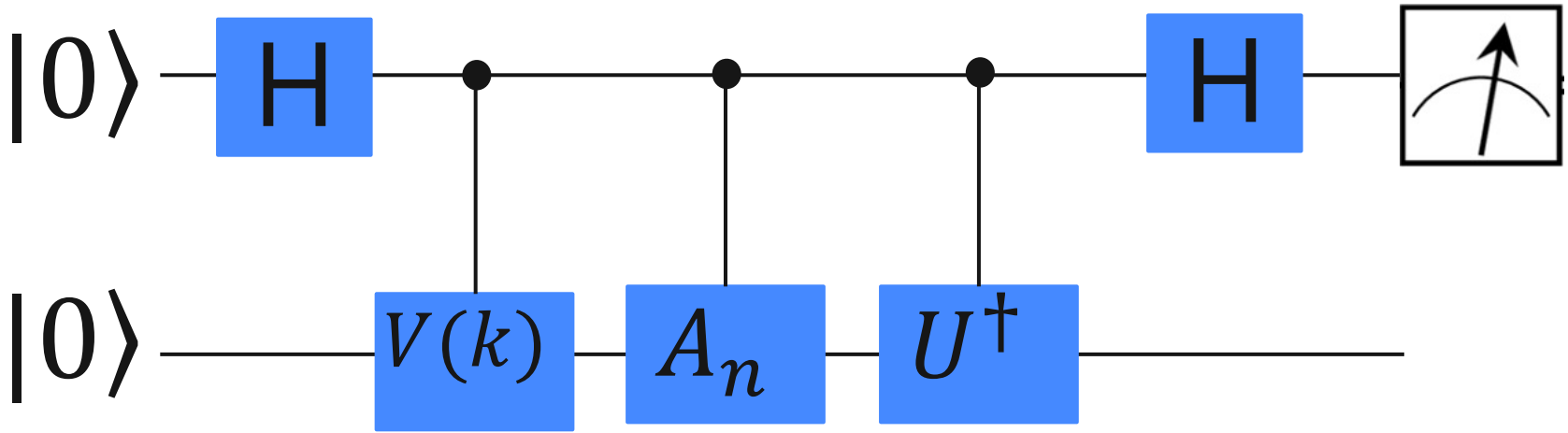
ユニタリ行列

「量子状態$|0\rangle$に対するユニタリ行列$U^\dagger A_n\, V(k)$の期待値」

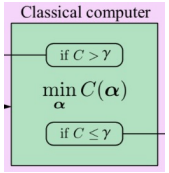
Hadamardテストによって各ユニタリ行列の期待値を個別に求めて、
積を計算し、係数をかけて和を取る！

$\langle 0|U^{\dagger}A_nV(k)|0\rangle$の値を得る量子回路

イメージはこのようになる



②コスト関数の求め方まとめ

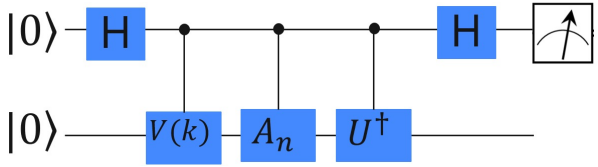


Quantum Tokyo

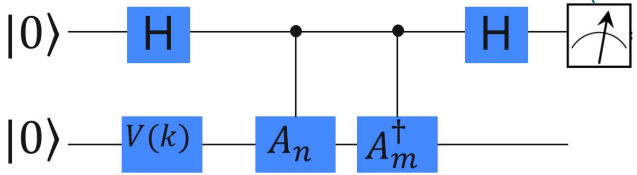
コスト関数に必要な計算方法をまとめると以下のようになっている。

$$|\langle b|\Phi(\mathrm{k})\rangle|^2 = \sum_m \sum_n c_m^* c_n \langle 0|U^\dagger A_n V(k)|0\rangle \langle 0|V^\dagger A_m^\dagger U|0\rangle$$

$$\widehat{C_P} = 1 - \frac{|\langle b|\Phi(k)\rangle|^2}{\langle \Phi(k)|\Phi(k)\rangle}$$

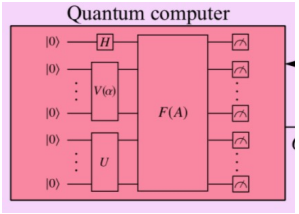


$$\langle \Phi(\mathrm{k})|\Phi(\mathrm{k})\rangle = \sum_m \sum_n c_m^* c_n \langle \psi(k)|A_m^\dagger A_n|\psi(k)\rangle$$



*実数のみの連立一次方程式の場合はユニタリ行列V、A、Uは実数で構成されるため、$\langle 0|V^\dagger A_m^\dagger U|0\rangle = \langle 0|U^\dagger A_n V(k)|0\rangle$となるので、同じ回路で算出可能である。
Notebookのデモではその仮定を置いて実装している。

③$V(k)$の構成



今回のデモでは3qubitを前提とし、以下の回路を採用している。

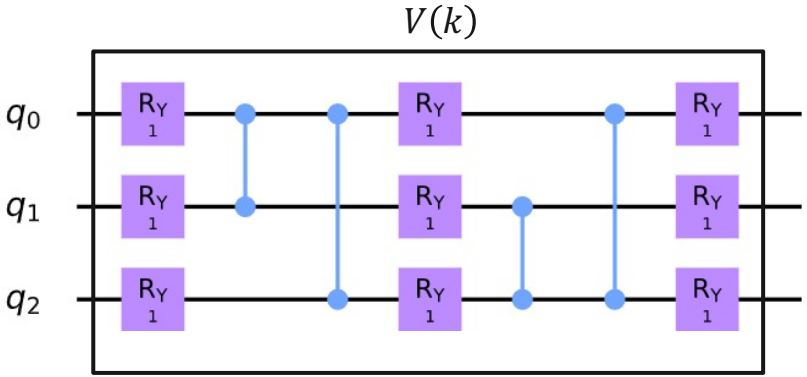
本編内ではFixed hardware Ansatzという名称としている。

回路の構造自体が固定で、変分パラメーターだけが調整される。

```
def apply_fixed_ansatz(qubits, parameters):

    for iz in range (0, len(qubits)):
        circ.ry(parameters[0][iz], qubits[iz])

    circ.cz(qubits[0], qubits[1])
    circ.cz(qubits[2], qubits[0])

    for iz in range (0, len(qubits)):
        circ.ry(parameters[1][iz], qubits[iz])

    circ.cz(qubits[1], qubits[2])
    circ.cz(qubits[2], qubits[0])

    for iz in range (0, len(qubits)):
        circ.ry(parameters[2][iz], qubits[iz])

circ = QuantumCircuit(3)
apply_fixed_ansatz([0, 1, 2], [[1, 1, 1], [1, 1, 1], [1, 1, 1]])
circ.draw(output='mpl')
```

$V(k)$

変分パラメーターk



全てのパラメーターの値が全て1となっている例

$V(k)|0\rangle = |\psi(k)\rangle$

kを調整

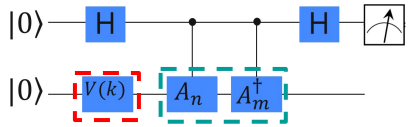
$\frac{|x\rangle}{|x|}$

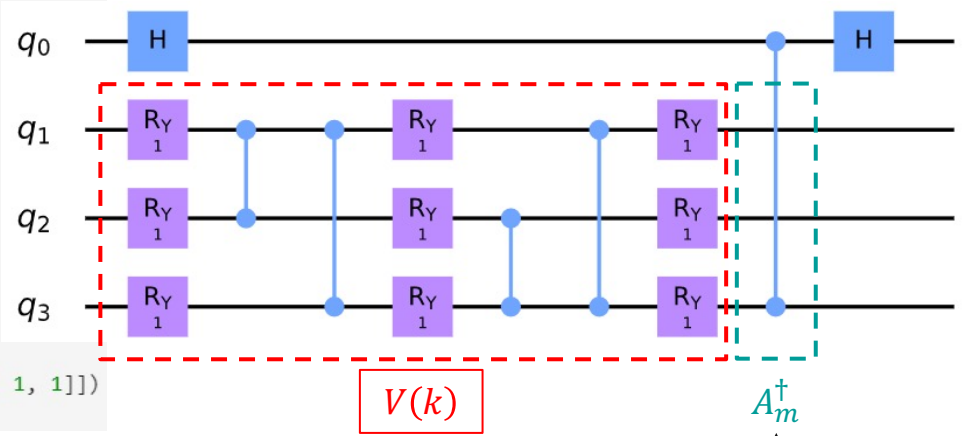
実機デモを行う前提

- 3qubit、つまり8変数の連立一次方程式を扱う
- $U = H \otimes H \otimes H$ に固定する
  - つまり$|b\rangle = H \otimes H \otimes H \, |000\rangle = (\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}},)$を扱う
- 係数は全て実数のみとする
  - つまり行列Aの成分は全て実数の場合を扱う
- 行列AはZとIのテンソル積で構成されたユニタリ行列の線形和のみを扱う
  - コードでは、“ゲートタイプ0”をI、”ゲートタイプ1”をZとして指定する
- 初期変分パラメーターは$\frac{k}{1000}$ $(k \in [0,3000])$でランダムに選択
- 変分パラメーターの最適化にはCOBYLA（ constrained optimization by linear approximation method ）法を使う

実機デモでのコスト関数の分母を求めるHadamardテストの回路の例

```
def had_test(gate_type, qubits, ancilla_index, parameters):

    circ.h(ancilla_index)

    apply_fixed_ansatz(qubits, parameters)

    for ie in range (0, len(gate_type[0])):
        if (gate_type[0][ie] == 1):
            circ.cz(ancilla_index, qubits[ie])

    for ie in range (0, len(gate_type[1])):
        if (gate_type[1][ie] == 1):
            circ.cz(ancilla_index, qubits[ie])

    circ.h(ancilla_index)

circ = QuantumCircuit(4)
had_test([[0, 0, 0], [0, 0, 1]], [1, 2, 3], 0, [[1, 1, 1], [1, 1, 1], [1, 1, 1]])
circ.draw(output='mpl')
```

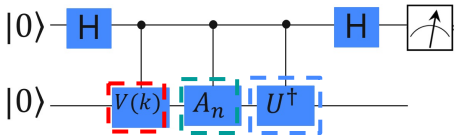




$A_n = I \otimes I \otimes I$

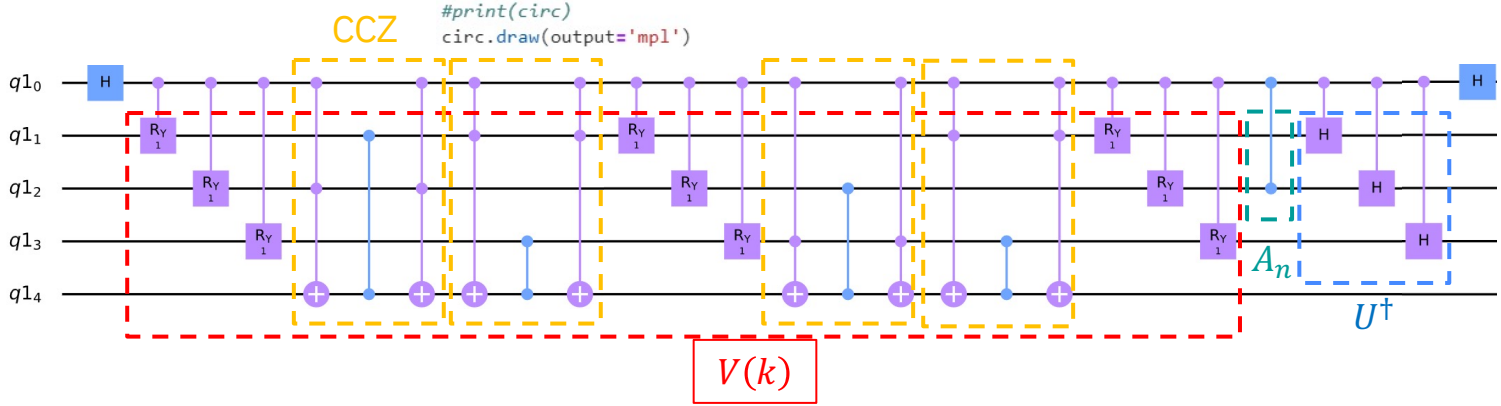
$A_m^\dagger = I \otimes I \otimes Z$

実機デモでのコスト関数の分子を求める Hadamardテストの回路の例




```
def special_had_test(gate_type, qubits, ancilla_index, parameters, reg):

    circ.h(ancilla_index)

    control_fixed_ansatz(qubits, parameters, ancilla_index, reg)
    for ty in range (0, len(gate_type)):
        if (gate_type[ty] == 1):
            circ.cz(ancilla_index, qubits[ty])


    control_b(ancilla_index, qubits)

    circ.h(ancilla_index)

q_reg = QuantumRegister(5)
circ = QuantumCircuit(q_reg)
special_had_test([0, 0, 1], [1, 2, 3], 0, [[1, 1, 1], [1, 1, 1], [1, 1, 1]], q_reg)
#print(circ)
circ.draw(output='mpl')
```

*textbook本編のコードには不備がありましたので、こちらは修正しています。後ほど本家のissueに上げておきます。後述のデモは正しいコードで稼働しています。



デモケース

① $A = 0.45Z_3 + 0.55I$

② $A = 0.55I + 0.225Z_2 + 0.225Z_3$

ここからは、本編のJupyter Notebookを参照

読んでいて出てきた疑問点

- 実際に利用することを想定すると、行列Aをユニタリ行列の線形結合を得た後でないと、VQLSを使えないのでは？
  - 前処理として、「行列Aをユニタリ行列の線形結合」を求める古典の処理が必要になるはず
- 一般のn変数連立一次方程式の場合のFixed Hardware Ansatzをどのように構成するべきか？
  - 3qubitでも他の構成方法を考えることは可能なはずだが、どのような点に注意して構成すべきかが不明。
  - n変数の場合についてもTextbookにはコメント無し。原論文からも読み取れず。
- HHLと異なり、スパース（行列の大半の成分が0）でなくても性能は安定するか？
  - 特に言及はされていない。
- 計算量的に、古典よりも本当に効率的か？
  - 少なくとも（Hadmardテストに必要なサンプリング数）×（線形分解した行列の種類）に比例するはず。
  - 計算量に関する議論は見つかっていない。

まとめ

- VQLSは連立一次方程式を、変分法を使ってパラメーターを調整して解を求める
- 量子状態の近さをコスト関数として定義する
- コスト関数は、ある状態に対するユニタリ行列の期待値として算出できる
- その期待値はHadamardテストを使って求める
- それらを古典の変分法を使って最適なパラメーターを求め、それを近似解とする

参考情報

- （原論文）**Variational Quantum Linear Solver**

  https://arxiv.org/abs/1909.05820

Thank you

Ayumu Shiraishi

AHA03784@jp.ibm.com