「単一システム　3．コード例」

2023/4/5

本日のSpeaker

横松　大作　（よこまつ・だいさく）

《仕　事》
・都内のユーザ系ＩＴ子会社にて
工業燃焼分野の数値シミュレーション、企業の基幹システム
の維持管理などを経て、基幹システムのプロジェクト支援に従事

《Qiskit Communityとのかかわり》

・2022年、IBMナレッジモール研究で量子コンピュータの活用研究に参加。

・1年ほど前、IBM Quantum Developer Certificate試験に合格。

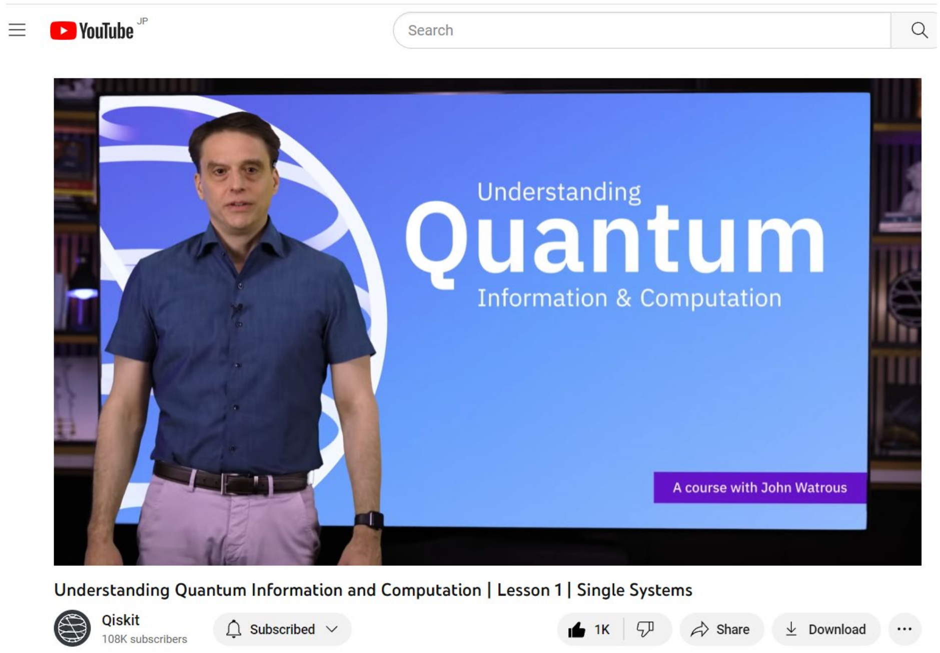
・昨年末、YouTube QiskitチャンネルでDr. John Watrousによる講義が公開されたのを見て、こちらのビデオとテキストを勉強したいと考え、Quantum Tokyoの勉強会に参加。

本日の内容

３．コード例

3.1　Pythonでのベクトルと行列

3.2　Qiskitでの状態、測定、および操作



https://www.youtube.com/watch?v=3-c4xJa7Flk

前々回、前回の内容はQiskit YouTubeのLesson 1 の講義内容にあたる。
本日の内容は、YouTubeの講義ビデオでは触れられていない。

前回の振り返り

量子状態の操作

線形マッピング　y = A x
量子状態ベクトルの操作はユニタリー行列で表される。
操作をすることはユニタリ—行列の積で表される。

コードとしては、行列×行列、行列×ベクトルについて考える

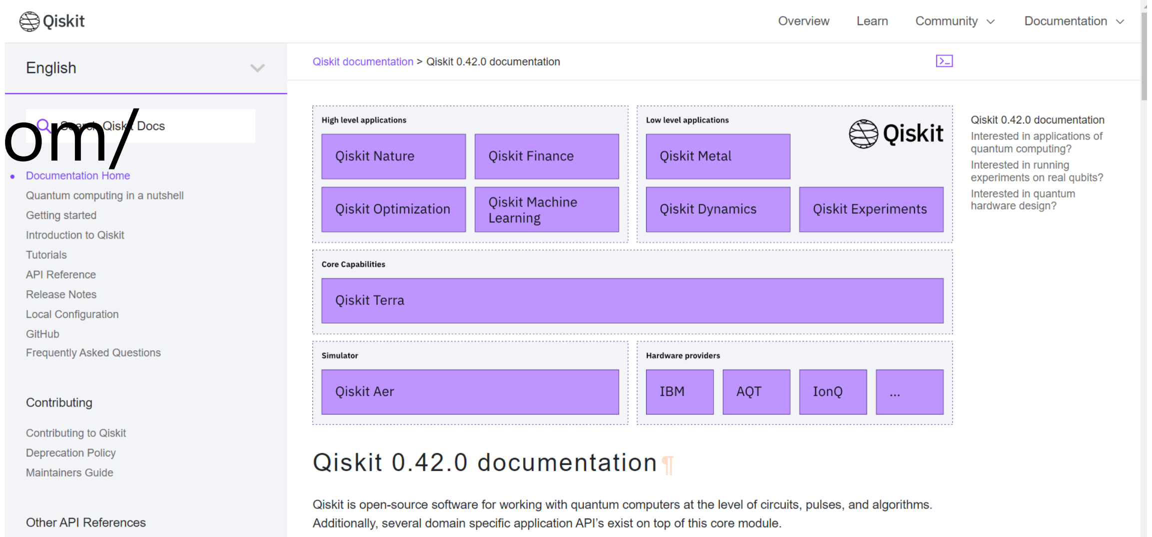
Qiskitについて

- Quantum information science kit, or Qiskit
- オープンソースの量子コンピューティングのSDK

- QiskitはPython言語で動作する

- IBM Quantum Platform
  https://quantum-computing.ibm.com/

- 自分のPCのPython環境に
  インストール



https://qiskit.org/documentation/index.html

3.1 Python でのベクトルと行列

NumPy ライブラリーの array クラスを使用し、
行列とベクトルの計算を実行

例：ベクトルのスカラー倍と和

```
from numpy import array

ket0 = array([1, 0])
ket1 = array([0, 1])

display(ket0/2 + ket1/2)
```

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\frac{1}{2}\left(|0\rangle + |1\rangle\right)$$

平均のベクトルを計算

Run

3.1 Python でのベクトルと行列

例：ベクトルのスカラー倍と和

```python
from numpy import array

ket0 = array([1, 0])
ket1 = array([0, 1])

display(ket0/2 + ket1/2)
```

Run

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\frac{1}{2}\left(|0\rangle + |1\rangle\right)$$

```
array([0.5, 0.5])
```

計算結果 $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$

3.1 Python でのベクトルと行列

例：ベクトルのスカラー倍と和

```
from numpy import array

ket0 = array([1, 0])
ket1 = array([0, 1])

display(ket0/2 + ket1/2)
```

Run

$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$\frac{1}{2}(|0\rangle + |1\rangle)$

明示的にdisplayコマンドを使わなくてもよい

```
array([0.5, 0.5])
```

計算結果 $\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$

3.1 Pythonでのベクトルと行列

Qiskit Textbookのコードセルについての注意

・特定のページで連続して実行すると累積的な効果がある
→あるコードセルで定義した内容はそれ以降のセルでも有効

・ページのリロード、別のページに切り替え→初期状態にリセット

・番号が付けられた各サブセクション内のコードセルは順番に実行されることを意図している
→エラーが出た場合は、サブセクション内の先頭のセルから実行

3.1 Python でのベクトルと行列

例：行列の和

```
M1 = array([ [1, 1], [0, 0] ])
M2 = array([ [1, 1], [1, 0] ])

M1/2 + M2/2
```

Run

$$M_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\tfrac{1}{2} M_1 + \tfrac{1}{2} M_2$$

```
array([[1. , 1. ],
       [0.5, 0. ]])
```

$$\tfrac{1}{2} M_1 + \tfrac{1}{2} M_2 = \begin{pmatrix} 1 & 1 \\ 0.5 & 0 \end{pmatrix}$$

3.1 Python でのベクトルと行列

例：行列とベクトルの積、行列の積

```python
from numpy import matmul

display(matmul(M1,ket1))
display(matmul(M1,M2))
display(matmul(M2,M1))
```

Run

$$M_1|1\rangle = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$M_1 M_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$M_2 M_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

```
array([1, 0])
array([[2, 1],
       [0, 0]])
array([[1, 1],
       [1, 1]])
```

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

3.2 Qiskitでの状態、測定、および操作

Qiskit には、状態、測定、および操作を簡単に作成および操作できるいくつかのクラスが含まれている。
ここでは、以下の例について示す：

・状態ベクトルの定義と表示
・Statevector を使用した測定のシミュレーション
・Operator と Statevector を使用した演算の実行
・量子回路の先を見据えて・・・量子回路の例

3.2 Qiskitでの状態、測定、および操作

例：Statevector クラスを使った状態ベクトルの定義

```python
from qiskit.quantum_info import Statevector
from numpy import sqrt

u = Statevector([1/sqrt(2), 1/sqrt(2)])
v = Statevector([(1+2.j)/3, -2/3])
w = Statevector([1/3, 2/3])

print("State vectors u, v, and w have been defined.")
```

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} \frac{1+2i}{3} \\ -\frac{2}{3} \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \end{pmatrix}$$

Run

```
State vectors u, v, and w have been defined.
```

3.2 Qiskitでの状態、測定、および操作

例：Statevector クラスを使った状態ベクトルの表示

```
display(u.draw('latex'))
display(v.draw('text'))
```

状態ベクトルの数式での表示

状態ベクトルの数値での表示

Run

$$\frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

```
[ 0.33333333+0.66666667j,-0.66666667+0.j        ]
```

$$\begin{pmatrix} \frac{1+2i}{3} \\ -\frac{2}{3} \end{pmatrix}$$

3.2 Qiskitでの状態、測定、および操作

例：量子状態ベクトルであるかどうかのチェック（is_validメソッド）
ユークリッドノルムで 1 であればTrue、そうでなければFalseを返す

```
display(u.is_valid())
display(v.is_valid())
display(w.is_valid())
```

Run

```
True
True
False
```

$$\sqrt{\left|\frac{1}{\sqrt{2}}\right|^2+\left|\frac{1}{\sqrt{2}}\right|^2}=1$$

$$\sqrt{\left|\frac{1+2i}{3}\right|^2+\left|-\frac{2}{3}\right|^2}=\sqrt{\frac{5}{9}+\frac{4}{9}}=1$$

$$\sqrt{\left|\frac{1}{3}\right|^2+\left|\frac{2}{3}\right|^2}=\sqrt{\frac{5}{9}}\neq 1$$

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

```
v = Statevector([(1+2.j)/3, -2/3])
v.draw('latex')
```

Run

$$(\frac{1}{3}+\frac{2i}{3})|0\rangle - \frac{2}{3}|1\rangle$$

測定する状態ベクトル

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

```
v.measure()
```

← Measureメソッドによる測定

Run

```
('1',
 Statevector([ 0.+0.j, -1.+0.j],
             dims=(2,)))
```

← 測定結果は‘１’

← 測定後の量子状態 $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$

```
('0',
 Statevector([0.4472136+0.89442719j, 0.        +0.j        ],
             dims=(2,)))
```

← 測定結果は‘0’

← 測定後の量子状態 $\begin{pmatrix} \frac{1+2i}{\sqrt{5}} \\ 0 \end{pmatrix}$

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

射影測定　Projection Operator を使って量子状態を測定する

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,\ \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1$$

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$Pr(0) = \langle\psi|{M_0}^\dagger M_0|\psi\rangle = \alpha^*\alpha\langle 0|0\rangle = |\alpha|^2$$ ：測定結果‘0’となる確率

$$Pr(1) = \langle\psi|{M_1}^\dagger M_1|\psi\rangle = \beta^*\beta\langle 1|1\rangle = |\beta|^2$$ ：測定結果‘1’となる確率

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

測定後の量子状態：

$$\frac{M_0|\psi\rangle}{\sqrt{Pr(0)}} = \frac{\alpha}{\sqrt{|\alpha|^2}}|0\rangle = \frac{\alpha}{|\alpha|}|0\rangle$$

$$\frac{M_1|\psi\rangle}{\sqrt{Pr(1)}} = \frac{\beta}{\sqrt{|\beta|^2}}|1\rangle = \frac{\beta}{|\beta|}|1\rangle$$

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

・測定結果　‘０’のときの測定後の量子状態

$$\frac{M_0|\psi\rangle}{\sqrt{Pr(0)}} = \frac{\alpha}{\sqrt{|\alpha|^2}}|0\rangle = \frac{\alpha}{|\alpha|}|0\rangle$$

$\alpha = \frac{1}{3} + \frac{2}{3}i$ として計算すると、$|\alpha| = \sqrt{\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2} = \sqrt{\frac{5}{9}}$

$$\frac{\alpha}{|\alpha|}|0\rangle = \sqrt{\frac{9}{5}}\begin{pmatrix} \frac{1}{3} + \frac{2}{3}i \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{5}}(1+2i) \\ 0 \end{pmatrix} = \frac{1}{\sqrt{5}}(1+2i)|0\rangle$$

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

・測定結果　‘1’のときの測定後の量子状態

$$\frac{M_1|\psi\rangle}{\sqrt{Pr(1)}} = \frac{\beta}{\sqrt{|\beta|^2}}|1\rangle = \frac{\beta}{|\beta|}|1\rangle$$

$\beta = -\frac{2}{3}$　として計算すると、$|\beta| = \sqrt{\left(-\frac{2}{3}\right)^2} = \sqrt{\frac{4}{9}}$

$$\frac{\beta}{|\beta|}|1\rangle = \sqrt{\frac{9}{4}}\begin{pmatrix} 0 \\ -\frac{2}{3} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1\rangle$$

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

・測定結果　‘0’のときの測定後の量子状態：$\begin{pmatrix} \frac{1+2i}{\sqrt{5}} \\ 0 \end{pmatrix} = \frac{1+2i}{\sqrt{5}}|0\rangle$

→ $|0\rangle$ と同等

・測定結果　‘1’のときの測定後の量子状態：$\begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1\rangle$

→ $|1\rangle$ と同等

・量子状態 $|\psi\rangle$ 、$|\phi\rangle$ について $|\psi\rangle = \alpha|\phi\rangle, \ |\alpha| = 1$
であるとき $|\psi\rangle$ 、$|\phi\rangle$ はグローバル位相が異なるという。
$\alpha = e^{i\theta}, \ \theta \in \mathbb{R}$ であるから、単位円上の複素数を掛けたものに等しい

3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

sample_counts メソッド：システム上の任意の回数の測定

```python
from qiskit.visualization import plot_histogram

statistics = v.sample_counts(1000)
display(statistics)
plot_histogram(statistics)
```

Run

1000回 測定を行う

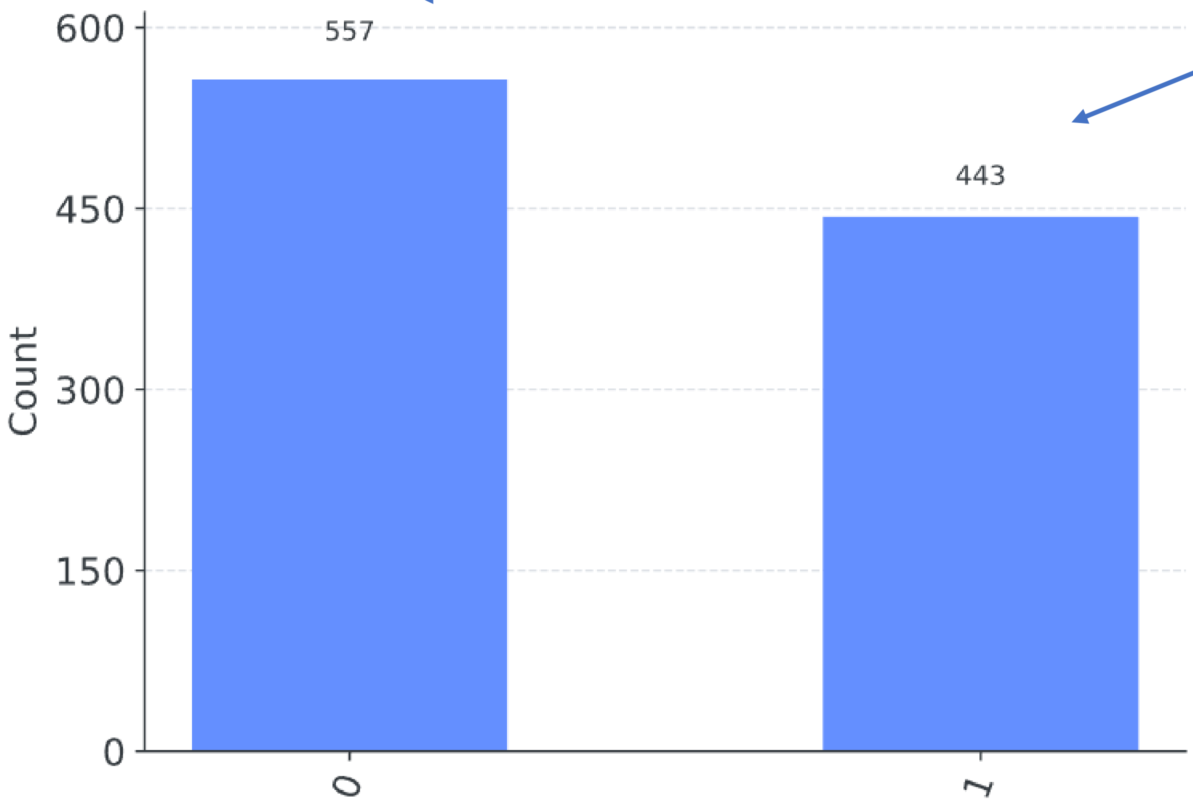
3.2 Qiskitでの状態、測定、および操作

例：Statevector を使用した測定のシミュレーション

sample_counts メソッド：システム上の任意の回数の測定

{'0': 557, '1': 443}

'0'、'1'を測定した回数



$$\left|\frac{1+2i}{3}\right|^2 = \frac{5}{9} \simeq 0.5556$$

$$\left|-\frac{2}{3}\right|^2 = \frac{4}{9} \simeq 0.4444$$

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

```
from qiskit.quantum_info import Operator

X = Operator([ [0,1],[1,0] ])
Y = Operator([ [0,-1.j],[1.j,0] ])
Z = Operator([ [1,0],[0,-1] ])
H = Operator([ [1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)] ])
S = Operator([ [1,0],[0,1.j] ])
T = Operator([ [1,0],[0,(1+1.j)/sqrt(2)] ])

v = Statevector([1,0])

v = v.evolve(H)
v = v.evolve(T)
v = v.evolve(H)
v = v.evolve(T)
v = v.evolve(Z)

v.draw('latex')
```

Run

Operator を定義

Statevector を定義

Operator をStatevectorに作用させ、結果を表示

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

```
v.draw('latex')
```

Run

$$\left(\frac{686 \cdot 2^{\frac{47}{48}} \cdot 3^{\frac{155}{192}} \cdot 5^{\frac{13}{96}} \cdot 7^{\frac{37}{96}}}{10125} + \frac{\sqrt{2}i}{4}\right)|0\rangle + (-0.353553390593 + 0.146446609407i)|1\rangle$$

？？？

```
v.draw('text')
```

drawのオプションをtextに変更して表示しなおす

Run

```
[ 0.85355339+0.35355339j,-0.35355339+0.14644661j]
```

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

1量子ビットの回路なので、手計算でも検証してみる

$$T = \begin{pmatrix} 1 & 0 \\ 0 & t \end{pmatrix} \qquad t = e^{\frac{i\pi}{4}} = \frac{1}{\sqrt{2}}(1+i)$$

$$\begin{aligned} ZTHTH|0\rangle &= ZTHT\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= ZTH\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & t \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= ZTH\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ t \end{pmatrix} \end{aligned}$$

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

$$
\begin{aligned}
&= ZTH\frac{1}{\sqrt{2}}\begin{pmatrix}1\\ t\end{pmatrix}\\
&= ZT\frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ t\end{pmatrix}\\
&= Z\begin{pmatrix}1 & 0\\ 0 & t\end{pmatrix}\frac{1}{2}\begin{pmatrix}1+t\\ 1-t\end{pmatrix}\\
&= \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1+t\\ t(1-t)\end{pmatrix}
\end{aligned}
$$

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

$$= \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1+t\\ t(1-t)\end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix}1+t\\ t(t-1)\end{pmatrix}$$

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

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

$$\begin{aligned}
\frac{1}{2}(1+t) &= \frac{1}{2}\frac{1}{\sqrt{2}}\left(1+i+\sqrt{2}\right) \\
&= \frac{1}{2\sqrt{2}}\left(1+\sqrt{2}+i\right) \\
&= \left(\frac{\sqrt{2}}{4}+\frac{1}{2}\right)+\frac{\sqrt{2}}{4}i
\end{aligned}$$

$$\approx 0.85355 + 0.35355i$$

3.2 Qiskitでの状態、測定、および操作

例：Operator と Statevector を使用した演算の実行

$$
\begin{aligned}
\frac{1}{2}t(t-1) &= \frac{1}{2}\frac{1}{\sqrt{2}}(1+i)\frac{1}{\sqrt{2}}(1+i-\sqrt{2}) \\
&= \frac{1}{4}(1+i-\sqrt{2}+i+i^2-\sqrt{2}i) \\
&= \frac{1}{4}(-\sqrt{2}+2i-\sqrt{2}i) \\
&= \frac{\sqrt{2}}{4}\left\{-1+(\sqrt{2}-1)i\right\} \\
&\approx -0.35355+0.14645i
\end{aligned}
$$

3.2 Qiskitでの状態、測定、および操作

例：量子回路の先を見据えて　（量子回路はLesson 3 で学ぶ）

```python
from qiskit import QuantumCircuit

circuit = QuantumCircuit(1)

circuit.h(0)
circuit.t(0)
circuit.h(0)
circuit.t(0)
circuit.z(0)

circuit.draw()
```

Run

QuantumCircuitクラスを使って量子回路を定義

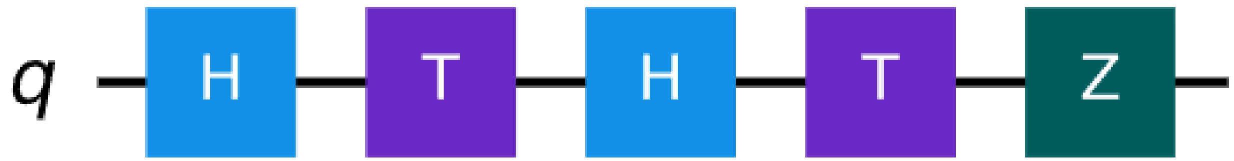
前項と同じゲートによる量子回路を定義

3.2 Qiskitでの状態、測定、および操作

例：量子回路の先を見据えて

```
circuit.draw()
```

Run



$$ZTHTH|0\rangle$$

初期状態

演算の順序：左から右へ

右から左へ

3.2 Qiskitでの状態、測定、および操作

例：量子回路の先を見据えて

```
ket0 = Statevector([1,0])
v = ket0.evolve(circuit)
v.draw('latex')
```

$$\begin{pmatrix}1\\0\end{pmatrix} = |0\rangle$$

circuitをket0に作用させる

Run

$$\left(\frac{686 \cdot 2^{\frac{47}{48}} \cdot 3^{\frac{155}{192}} \cdot 5^{\frac{13}{96}} \cdot 7^{\frac{37}{96}}}{10125} + \frac{\sqrt{2}i}{4}\right)|0\rangle + (-0.353553390593 + 0.146446609407i)|1\rangle$$

$$\approx (0.85355 + 0.35355i)\,|0\rangle + (-0.35355 + 0.14645i)\,|1\rangle$$

3.2 Qiskitでの状態、測定、および操作

例：量子回路の先を見据えて

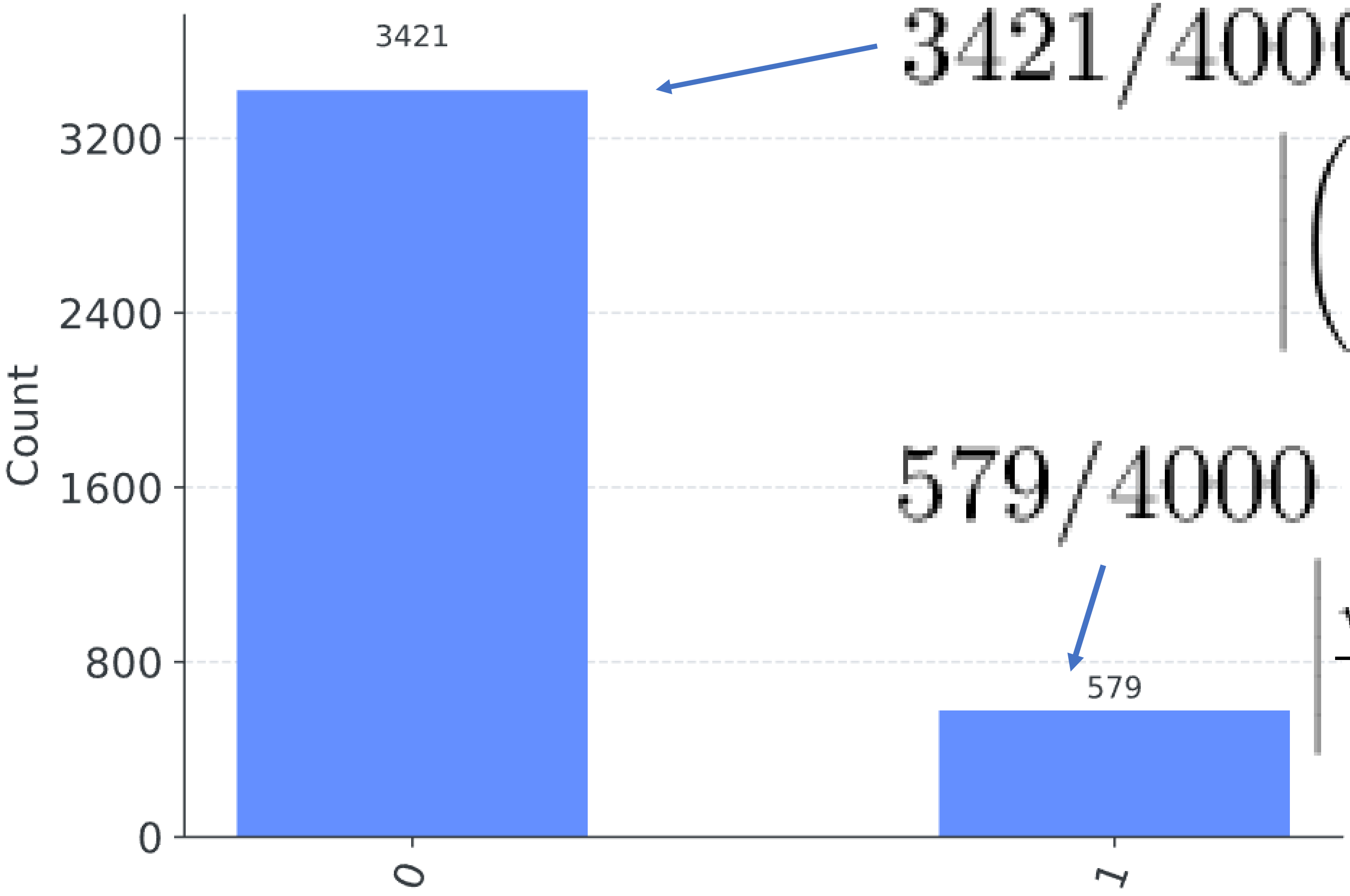
```
statistics = v.sample_counts(4000)
plot_histogram(statistics)
```

4000回実行、測定を繰り返す

Run

3.2 Qiskitでの状態、測定、および操作

例：量子回路の先を見据えて



$$3421/4000 = 0.85525$$

$$\left| \left(\frac{\sqrt{2}}{4} + \frac{1}{2} \right) + \frac{\sqrt{2}}{4} i \right|^2 \approx 0.85355$$

$$579/4000 = 0.14475$$

$$\left| \frac{\sqrt{2}}{4} \left\{ -1 + (\sqrt{2} - 1) i \right\} \right|^2 \approx 0.14645$$

まとめ

量子状態の操作

量子状態ベクトルの操作はユニタリー行列で表される。
操作をすることはユニタリ—行列の積で表される。

↓

Qiskitではコードとして、Statevector、Operator、QuantumCircuitなどが用意されており、これらを使ってユニタリー行列の積を表し、 シミュレータ上で測定、操作を行うことができる。

ご清聴ありがとうございました。おつかれさまでした。