

# Qiskit Textbook 9

## Estimating Pi Using Quantum Phase Estimation Algorithm

---

Ayumu Shiraishi

Qiskit Advocate

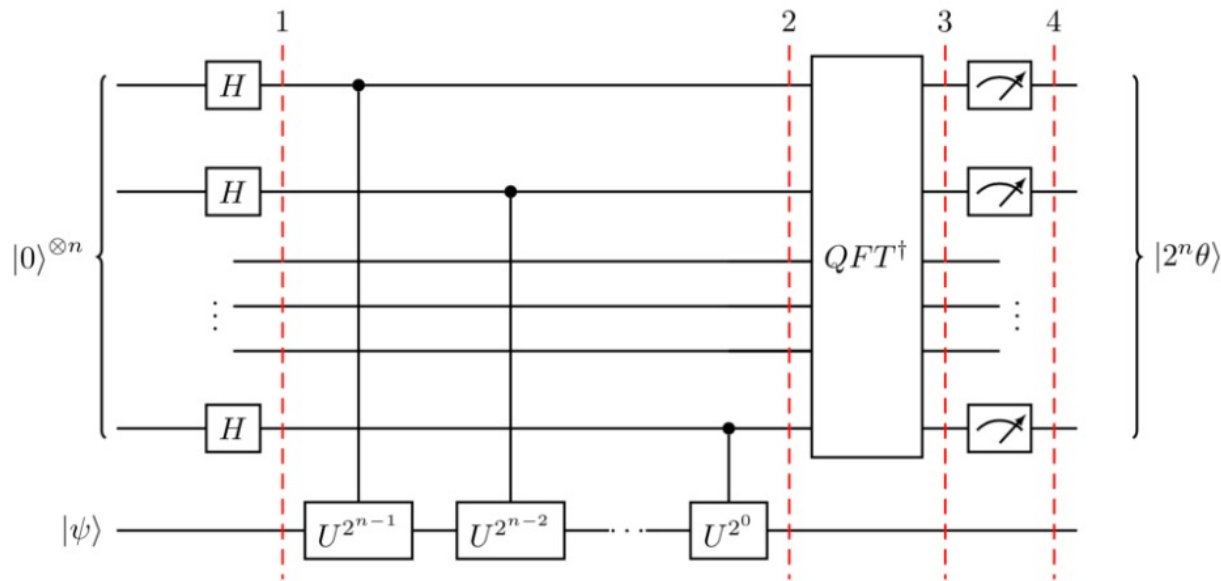


# 位相推定 (Quantum Phase Estimation) の復習

当デモでは、位相推定を使って円周率 $\pi$ を求めてみよう。

QPEの詳細はQiskit Textbook 3.8章を復習を！

[https://www.youtube.com/watch?v=y7t\\_A0WFl00&t=2014s](https://www.youtube.com/watch?v=y7t_A0WFl00&t=2014s)



# QPEを使うエッセンス

位相推定は、 $U|\psi\rangle = e^{i\varphi}|\psi\rangle$ の $\varphi$ の値を求めるアルゴリズム

位相推定の式は次のようにも書き直せる

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle \quad \text{ここで } \theta = 0.\theta_1\theta_2\cdots = \frac{\theta_1}{2^1} + \frac{\theta_2}{2^2} + \cdots$$

(ただし、 $\theta_i = 0$  or  $1$ の2進数の値のどちらかになる。)

このことから、

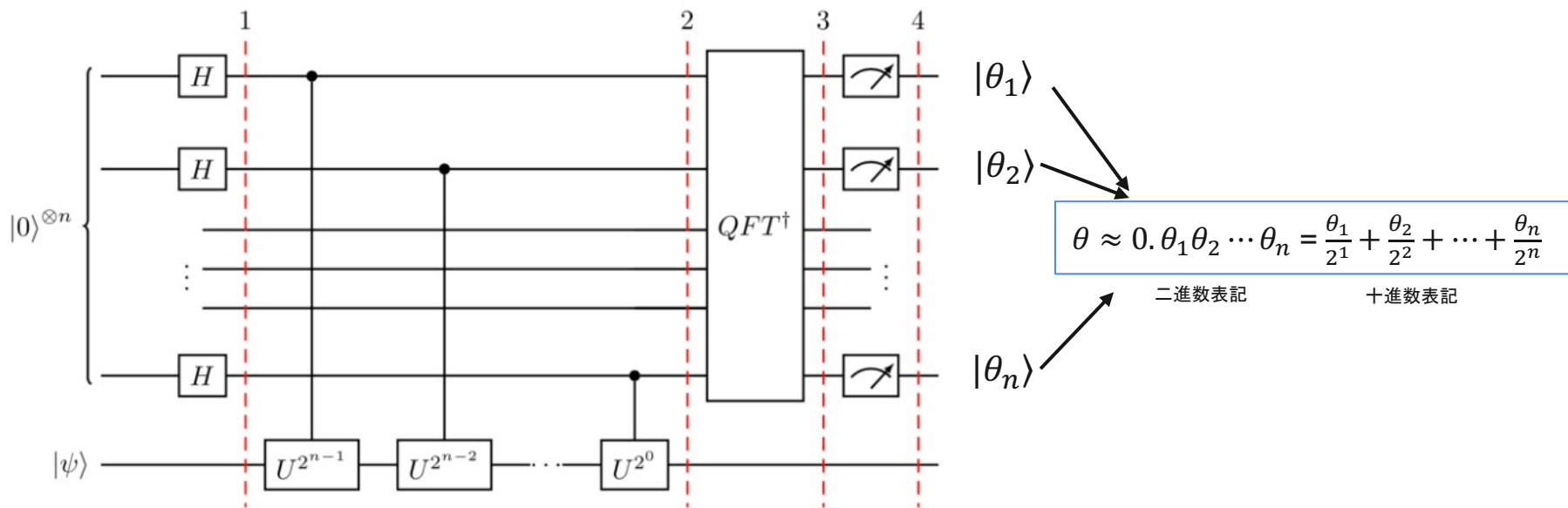
$$\varphi = 2\pi\theta \quad \Rightarrow \quad \pi = \frac{\varphi}{2\theta}$$

となる。

ここで、 $\varphi$ はこのデモにおいて実験者が任意に設定可能な値であり、 $\theta$ はQPEの観測によって測定可能な値である。

# QPEを使うエッセンス

QPEによって、量子ビット数の精度で位相を推定できる。



# デモの設定

$|\psi\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ 、 $U = U_1(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$ とすると、 $U|1\rangle = e^{i\varphi}|1\rangle$ とできる。

さらに、 $\varphi = 1$ と設定しておくともデモが構成しやすい。つまり、 $\pi = \frac{1}{2\theta}$ となる。

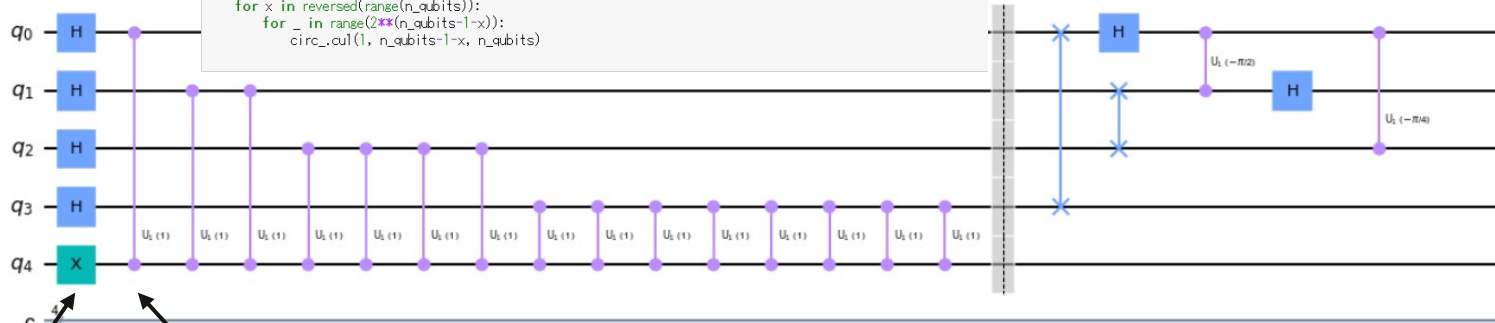
以上の状況で、 $\theta$ （の近似値）をQPEを構成して求める。

# 参考：実行時の回路 (n=4)

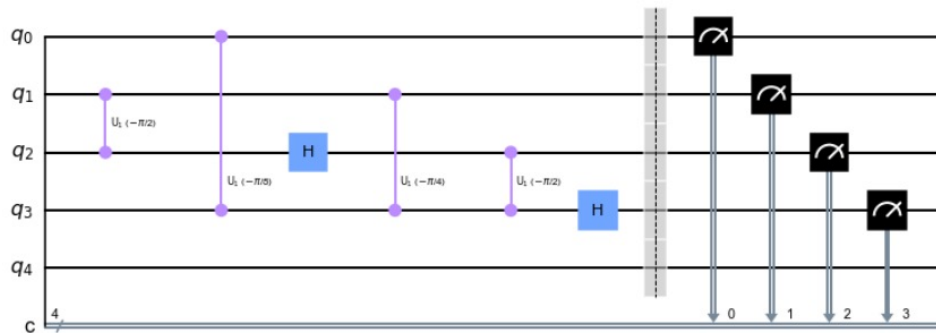
```
## 量子位相推定の初期状態のコード
## Qiskit textbook qiskit.org/textbookから抜粋
## 開始時の状態には最後の量子ビット以外にHを適用し、最後の量子ビットには $|\psi\rangle = |1\rangle$ にセットすることに注意
```

```
def qpe_pre(circ_, n_qubits):
    circ_.h(range(n_qubits))
    circ_.x(n_qubits)

    for x in reversed(range(n_qubits)):
        for _ in range(2**(n_qubits-1-x)):
            circ_.cnot(1, n_qubits-1-x, n_qubits)
```



$|1\rangle$ を作る  $\rightarrow$   $\rightarrow$  controlled - Uを作用



```
## 逆量子フーリエ変換のコード
## Qiskit Textbook qiskit.org/textbookから抜粋
```

```
def qft_dagger(circ_, n_qubits):
    """n-qubit QFTdagger the first n qubits in circ"""
    for qubit in range(int(n_qubits/2)):
        circ_.swap(qubit, n_qubits-qubit-1)
    for j in range(0, n_qubits):
        for m in range(j):
            circ_.cnot(-np.pi/float(2**(j-m)), m, j)
        circ_.h(j)
```

# 実行結果

Job Status: job has successfully run  
2 qubits,  $\pi \approx 2.0$

Job Status: job has successfully run  
3 qubits,  $\pi \approx 4.0$

Job Status: job has successfully run  
4 qubits,  $\pi \approx 2.6666666666666665$

Job Status: job has successfully run  
5 qubits,  $\pi \approx 3.2$

Job Status: job has successfully run  
6 qubits,  $\pi \approx 3.2$

Job Status: job has successfully run  
7 qubits,  $\pi \approx 3.2$

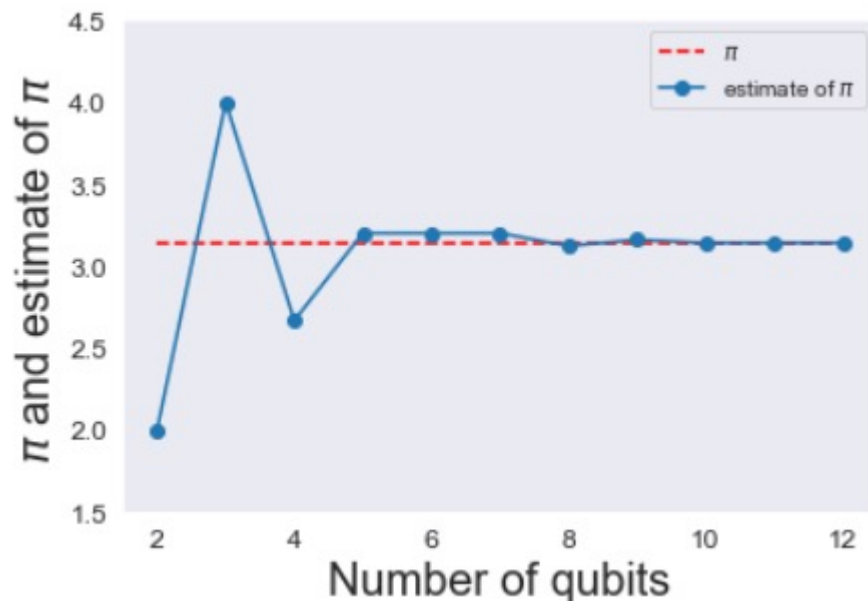
Job Status: job has successfully run  
8 qubits,  $\pi \approx 3.1219512195121952$

Job Status: job has successfully run  
9 qubits,  $\pi \approx 3.1604938271604937$

Job Status: job has successfully run  
10 qubits,  $\pi \approx 3.1411042944785277$

Job Status: job has successfully run  
11 qubits,  $\pi \approx 3.1411042944785277$

Job Status: job has successfully run  
12 qubits,  $\pi \approx 3.1411042944785277$



# Thank you

Ayumu Shiraishi

AHA03784@jp.ibm.com

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).