Qiskit Tutorial Finance
QAE と European Call Option Pricing

Kenji Tanaka
IT architect

お前誰やねん

田中健之 (Tanaka Kenji)

IT architect

- OOP, SOA, BPM, EA, RUP, Agile, Lean
- 提案・開発・保守、挑戦・失敗・火消
- 子供にはわかってもらえない辛い仕事

大学院の専攻は応用物理

- 超音波とプラズマの実験をしていました

いきさつ

- 去年勉強会で発表した QAE が Finance Tutorial 解説に役立つので再放送します
    - ネタ変わってません
- Tutorial で使っているアルゴリズムが QAE から IterativeQAE に変わりました
    - QAE から学習するとわかりやすいので QAE のまま行きます!
- 量子コンピュータで金融計算は European Call Option が入門しやすい
    - 迷ったらこれをお勧めします

概要

- デリバティブの Option 価格計算
  - 最も有名な計算方法 Black Sholes モデル
    - 価格変化に正規分布（Log Normal）を仮定している
    - 複雑な分布や条件を適用するとすぐに、積分が解析的に解けなくなるため、数値計算が必要

- 解析的に解けない関数の数値積分
  - 現在は Monte Caro 法が主流だが、時間がかかる
  - 量子コンピュータの <u>QAE で積分計算をすると Quadratic に早い</u>

- 以下の論文を参考にしました
  - Quantum computational finance: Monte Carlo pricing of financial derivatives https://arxiv.org/abs/1805.00109

金融工学と量子計算

- 金融工学の様々な用途に、色々なアルゴリズムが提案されている
  - ポートフォリオ最適化、リスク計算、デリバティブ、・・・
  - HHL, QAE, QAOA, qGAN, ・・・

– 今回取り上げる QAE による Option Pricing はそのうちの一例に過ぎない
  - Qiskit Tutorials に金融分野だけでも色々あります
    – https://qiskit.org/documentation/tutorials/finance/index.html

Option について

Option とは金融派生商品（デリバティブ）の一種

- デリバティブとは
  - 株、債券、などを原資産とした派生商品
  - 原資産の価格変動で、デリバティブの価格（価値）も変化する

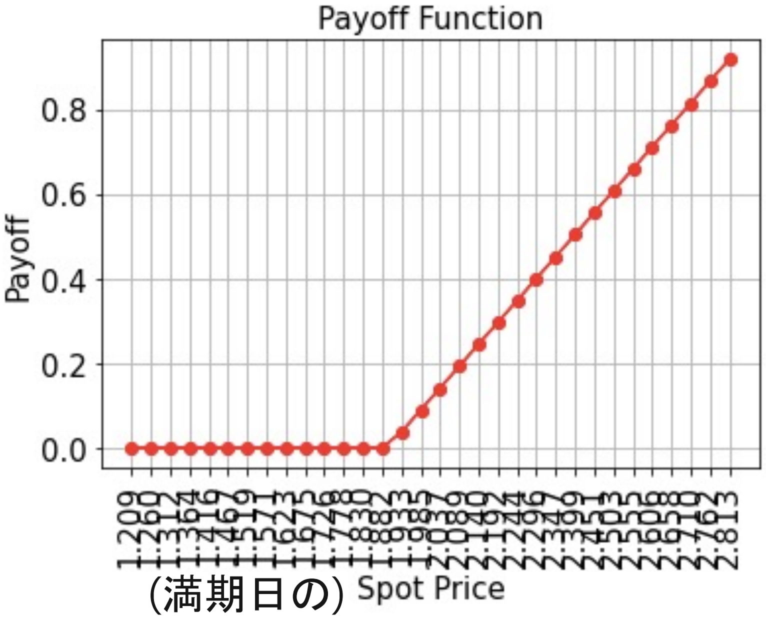
Option とは

- Maturity Date (満期日）に Strike Price (行使価格)で原資産を売買する権利
  - Call Option 買う権利 / Put Option 売る権利
- 様々な商品があります
  - European, American, Asian
  - 計算方法が若干違います。それぞれ論文や Tutorial があります
  - Option の組み合わせもあります

→ 今回は European Call Option を対象にします

European Call Option のペイオフ

- ペイオフダイアグラム
  - 損益採算を示したダイアグラム
- 満期日に原資産の価格が Strike Price よりも
  - 高値であれば、権利を行使して、利益を得る
  - 低ければ、権利行使しない　（損しない）
- Option はナンボあっても邪魔にはならない
  - 無料でもらえるならば、もらいましょう
  - <u>では Option をいくらならば買いますか？（売りますか？）</u>



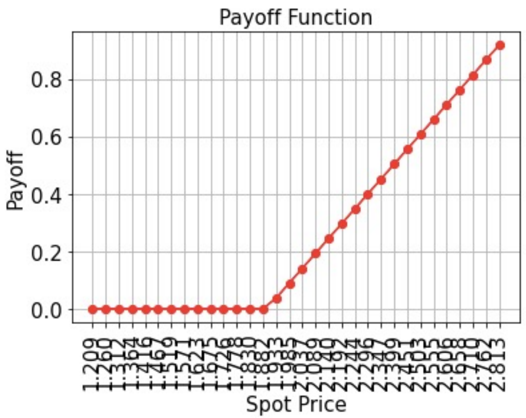
$$\max\{S_T - K, 0\}$$

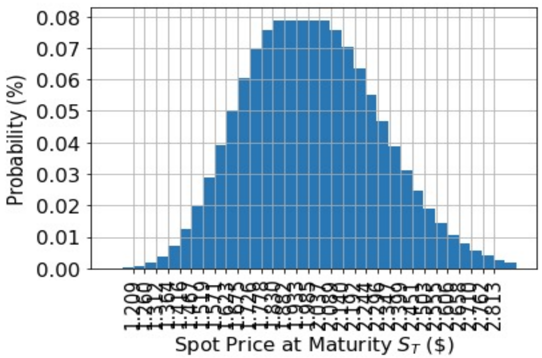
Option の価格

Option の価格/適正価格/理論値

Option の価値 ＝ Payoff の期待値

実現確率（確率分布）と Payoff をかけて積分する





但し

- Payoff は誰が計算しても同じ
- 実現確率は選択する

ブラックショールズモデル

$$C = SN(d_1) - Ke^{-rt}N(d_2)$$

$$d_1 = \frac{\ln(S/R) + \left(r + \sigma^2/2\right)t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

C	コールオプションの価格
S	原資産の価格
N	標準正規分布の累積分布関数
K	コール・オプションの行使価格
t	満期時点
r	満期における無リスク金利
e^-rt	割引係数
σ	原資産の収益率の標準偏差

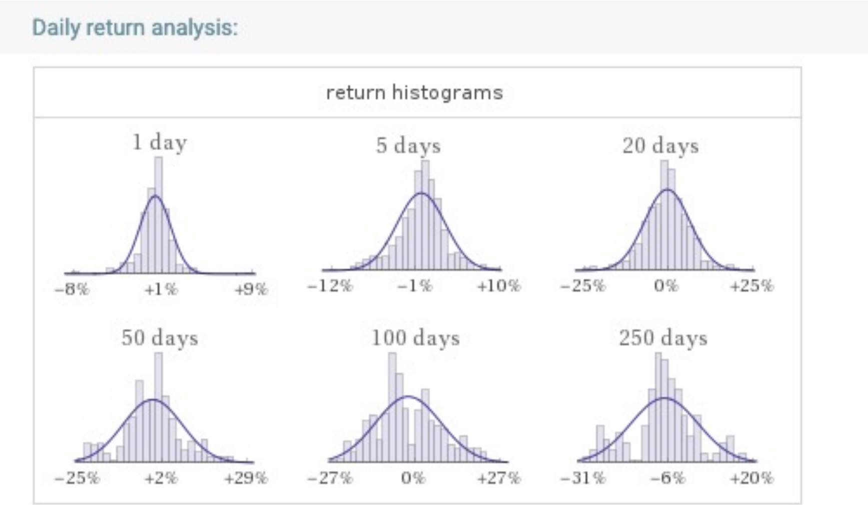
オプションの理論価格はブラックショールズモデルで与えられる

原資産の価格変動は Wiener 過程（ブラウン運動）

参考

https://www.sigmabase.co.jp/useful/keyword/h/bs_model.html

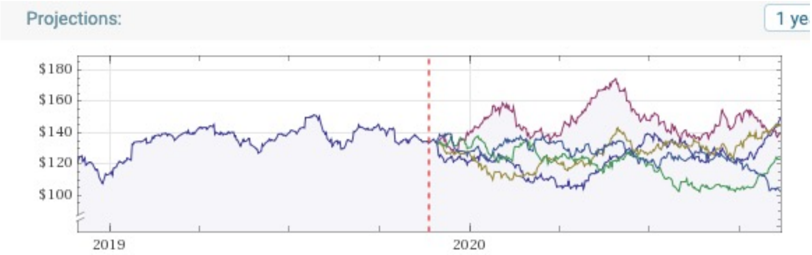
原資産の価格変動の例



株価は正規分布に従うと言えるか？

(あなたの選択です)

https://www.wolframalpha.com/



50% range, 1 month	$127.19 (-4.9%) to $139.02 (+3.9%)
95% range, 1 month	$116.86 (-12.6%) to $151.32 (+13.1%)
50% range, 1 year	$106.83 (-20.1%) to $145.30 (+8.6%)
95% range, 1 year	$79.70 (-40.4%) to $194.77 (+45.6%)

(log-normal random walks based on historical parameters)

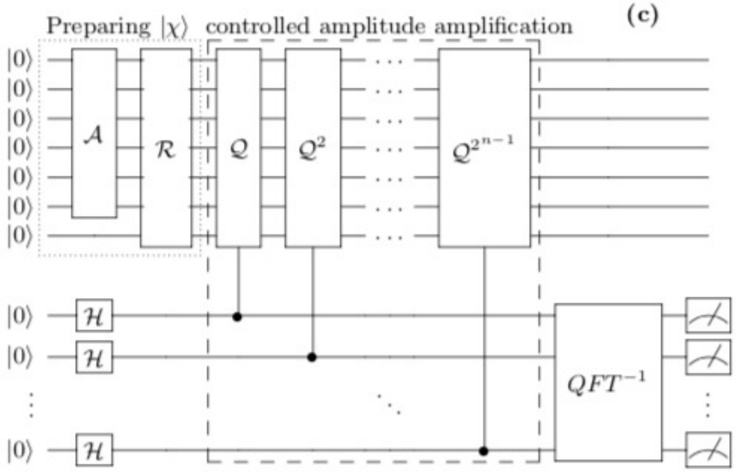
期待値の計算は、確率分布を

- 近似すれば Excel でも計算可能
- 複雑にすれば難しくなる

金融機関では難しい計算をしている(企業秘密)

- qGAN で学習しても良い

QAE: Quantum Amplitude Estimation



QAE で期待値を計算できる

- Shor の回路と似ている
- Q は Grover の回路

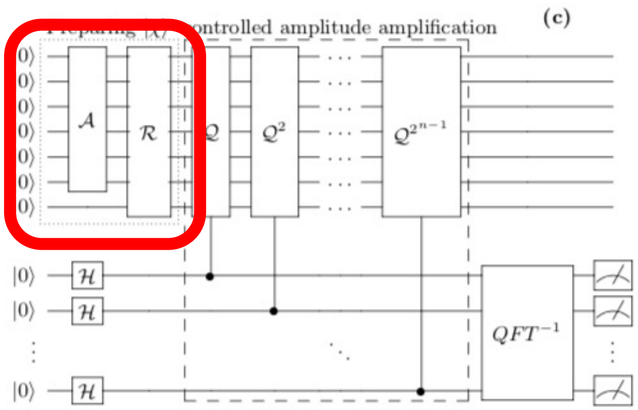
計算

- 求めたい期待値　μ
- Q の固有値 e^iθ
- とすると、次の関係がある

$$1 - 2\mu = \cos(\theta/2)$$

- そのため、QPE で θ を読み出すと期待値が計算できる

確率分布と Payoff を Qubit にロードする



A で確率分布を設定する (ここではLogNormal)

$$\mathcal{A}|0^n\rangle = \sum_{x=0}^{2^n-1} a_x |x\rangle$$

R の|1> に Payoff v(x) を設定

$$\mathcal{R}|x\rangle|0\rangle = |x\rangle\left(\sqrt{1-v(x)}|0\rangle + \sqrt{v(x)}|1\rangle\right)$$

この状態をχとおくと

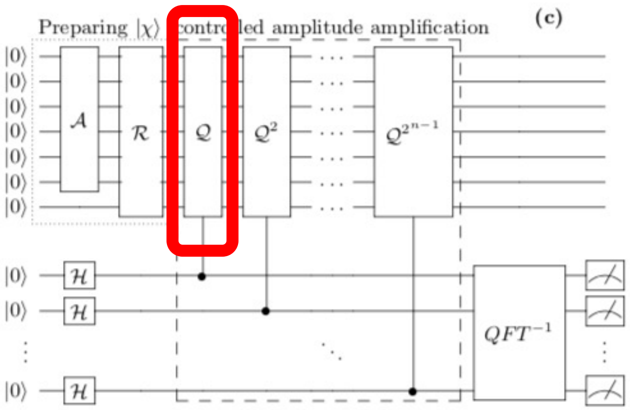
$$|\chi\rangle \equiv \mathcal{R}(\mathcal{A} \otimes \mathcal{I}_2)\left|0^{n+1}\right\rangle$$

期待値 μ は以下の Operator の期待値になる

$$\mu := \langle\chi|(\mathcal{I}_{2^n} \otimes |1\rangle\langle 1|)|\chi\rangle$$

→ 欲しい積分の値は、期待値 μ

Grover



Grover の回路を通すと、

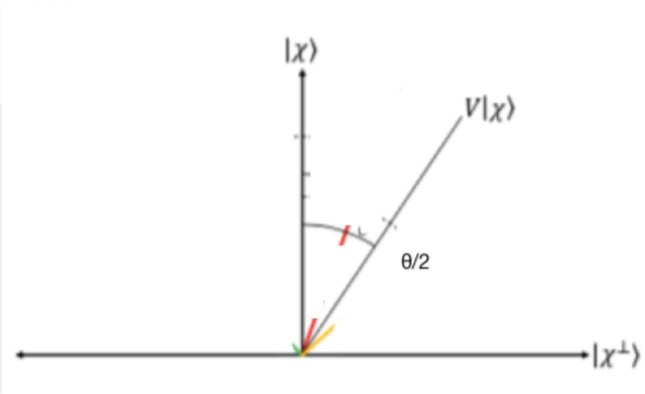
期待値μ と以下の関係になる固有値 e^iΘ が出てくる

$$1 - 2\mu = \cos(\theta/2)$$

Grover

v を以下のように定義する

$$\mathcal{V} := \mathcal{I}_{2^{n+1}} - 2\mathcal{I}_{2^n} \otimes |1\rangle\langle 1|$$



μ との関係は以下になる

$$\langle\chi|\mathcal{V}|\chi\rangle = 1 - 2\mu$$

（μの定義から）

$$\mu := \langle\chi|(\mathcal{I}_{2^n} \otimes |1\rangle\langle 1|)|\chi\rangle$$

θ を以下のようにおくと

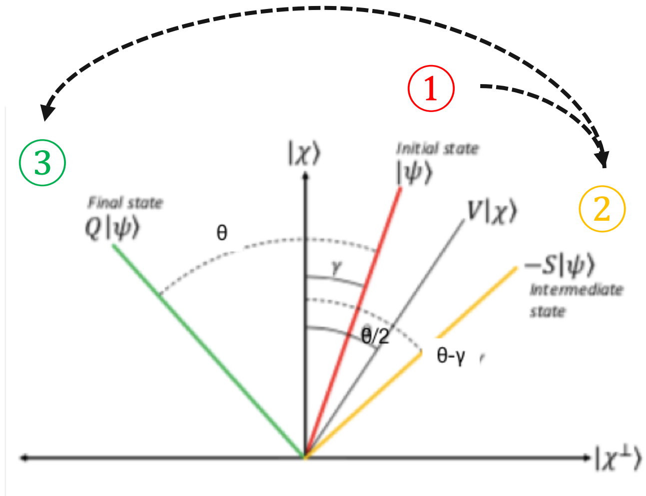
$$\mathcal{V}|\chi\rangle = \cos(\theta/2)|\chi\rangle + e^{i\phi}\sin(\theta/2)\left|\chi^{\perp}\right\rangle$$

μ と θ の関係は以下になる

$$1 - 2\mu = \cos(\theta/2)$$

→ θ を図のようにとると、θ から μ が計算できる

Grover



Grover でおなじみの回路で反転（v|χ> と |χ> を軸に任意の |ψ> を反転）

$$\mathcal{Q} := \mathcal{U}\mathcal{S}$$

$$\mathcal{U} := \mathcal{I}_{2^{n+1}} - 2|\chi\rangle\langle\chi|$$

$$\mathcal{S} := \mathcal{I}_{2^{n+1}} - 2\mathcal{V}|\chi\rangle\langle\chi|\mathcal{V}$$

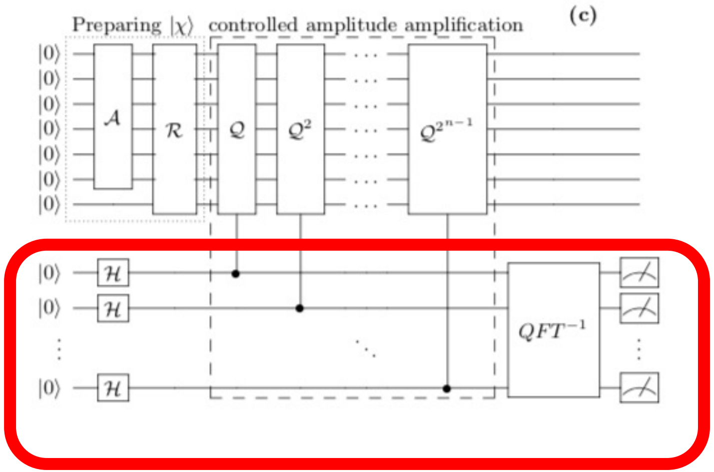
Qの固有値としてθが出てくる

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

固有値 $e^{\pm i\theta}$

http://www.f-denshi.com/000TokiwaJPN/05unitr/020unt.html

位相Θを QPE で読み出す



$$|\chi\rangle = \frac{1}{\sqrt{2}}(|\psi_+\rangle + |\psi_-\rangle)$$

$$Q|\chi\rangle = \frac{1}{\sqrt{2}}\left(e^{i\theta}|\psi_+\rangle + e^{-i\theta}|\psi_-\rangle\right)$$

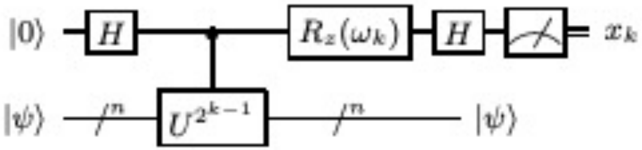
- QPE の入力は固有ベクタを用意する必要がある
- |χ> は Q の固有ベクタの重ね合わせのため、+- θ の 2 つを等確率で得る
- 固有値はどちらをとっても同じ結果になる

→ 位相 θ を QPE で読み出し、期待値 μ を算出する

demo

- Qiskit Documentation ヨーロピアン・コール・オプションの価格推定
  - https://qiskit.org/documentation/locale/ja_JP/tutorials/finance/03_european_call_option_pricing.html

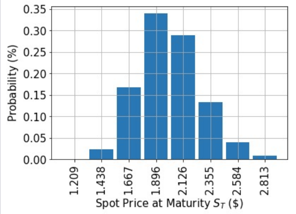
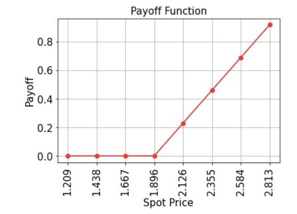
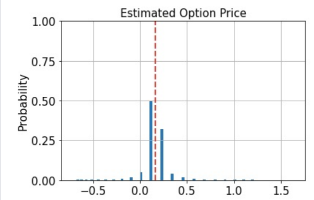
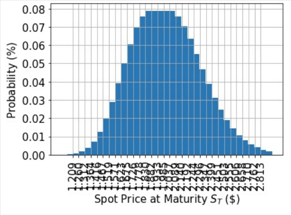
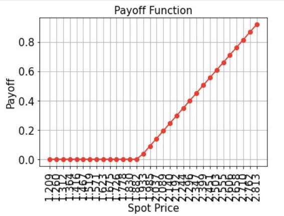
- Iterative QAE
  - 2020年まではこのチュートリアルは QAE でしたが、最新版は Iterative QAE に変わっています
  - 通常の QAE に比べ、ゲートが少なく、早くなっています
  - Iterative QAE は QPE を Iterative QPE にしたものですが、単純に置き換えたものではなく改良しているようです
    - Iterative Amplitude Estimation https://arxiv.org/abs/1912.05559
  - 詳しくは QPE と IQPE を比べると良いと思います



QPE の実行

本質的な比較ではありませんが、Qbit 数を増やすと遅くなりました

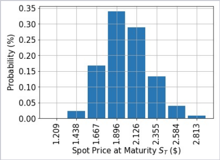
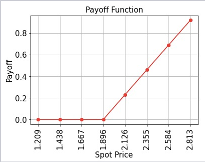
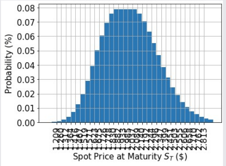
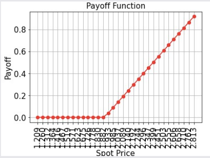
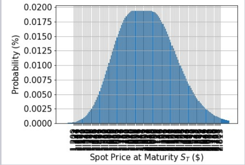
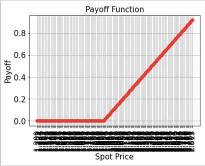
古いバージョンの Tutorial を MacBook Air 2015 で実行

Uncertainty	QPE		result	
3 qbit	6 qbit	 		2分
5 qbit	8 qbit	 		8 時間

Iterative QPE の実行

単純な比較はできませんが、Qbit 数を増やしても Iterative QPE ではあまり時間が変わりませんでした

最新の Tutorial を MacBook Air 2015 で実行

Uncertainty		result	
3 qbit	 	Exact: 0.1623 Estimated: 0.1710 C Interval: [0.1635, 0.1785]	1.2 秒
5 qbit	 	Exact: 0.1673 Estimated: 0.1755 C Interval: [0.1667, 0.1844]	3.3 秒
7 qbit	 	Exact: 0.1668 Estimated: 0.1690 C Interval: [0.1602, 0.1779]	15秒

まとめ

- Option Pricing を量子コンピュータで計算すると Quadratic に早くなる
  - 期待値の積分計算が早くなる

参考文献

- Quantum computational finance: Monte Carlo pricing of financial derivatives
  - https://arxiv.org/abs/1805.00109

- Quiskit Aqua Tutorial: European Call Option
  - https://qiskit.org/documentation/tutorials/finance/5_european_call_option_pricing.html

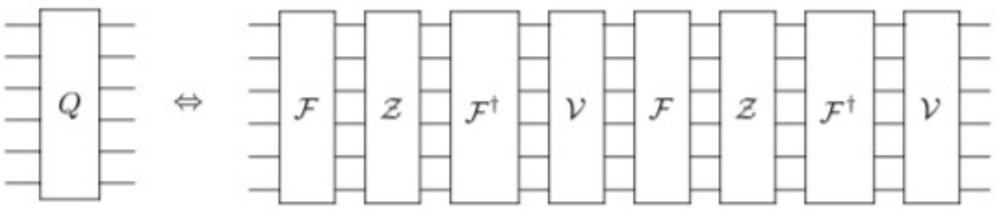
- 金融工学入門　第２版
  - https://www.amazon.co.jp/dp/4532134587/

– Iterative Quantum Amplitude Estimation
  - https://arxiv.org/abs/1912.05559

– Arbitrary accuracy iterative phase estimation algorithm as a two qubit benchmark (IQPE)
  - https://arxiv.org/abs/quant-ph/0610214

Grover 回路の構成

(a)

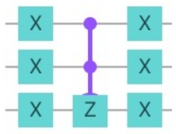


※ゲートの順番が逆ではないかと思うのですが、
元の論文のママです

$$\mathcal{F} := \mathcal{R}(\mathcal{A} \otimes \mathcal{I}_2)$$

$$\mathcal{V} := \mathcal{I}_{2^{n+1}} - 2\mathcal{I}_{2^n} \otimes |1\rangle\langle 1|$$

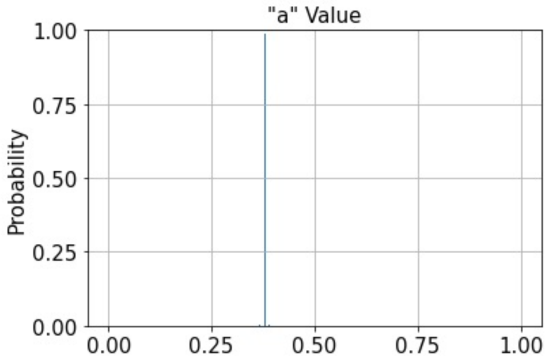
$$\mathcal{Z} := \mathcal{I}_{2^{n+1}} - 2\left|0^{n+1}\right\rangle\left\langle 0^{n+1}\right|$$

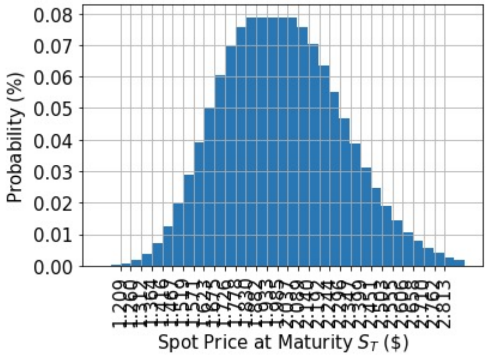


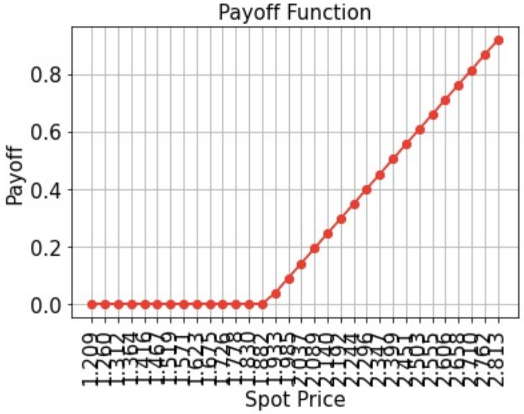
QPE Tutorial 高解像度化

Quantum Tokyo

- Qubit 数を増やして simulate した結果
    - Uncertainty qubits = 5
    - QPE qubits = 8
    - 計算 8時間以上 (Mac Book Air 2015)
        - CPU times: user 8h 10min 17s, sys: 48min 24s, total: 8h 58min 41s Wall time: 5h 49min 36s
        - 最初の設定（Uncert=3, QPE=6）では２分
        - CPU times: user 1min 28s, sys: 1.4 s, total: 1min 30s Wall time: 1min 44s







Iterative QPE Tutorial 高解像度化

２箇所変更する

num_uncertainty_qubits = 7

ae = IterativeAmplitudeEstimation(epsilon=epsilon, alpha=alpha,

state_preparation=european_call,

objective_qubits=[num_uncertainty_qubits],

post_processing=european_call_objective.post_processing)