ご購入はこちら

ダウンロード・データあります
http://www.cqpub.co.jp/interface/download/contents.htm

第3章 数十量子ビットを使う大規模な量子回路も試せる

Python対応高速シミュレータで乱数生成＆量子機械学習

藤井 啓祐

第3章では，Pythonでプログラミングできる量子コンピュータ・シミュレータを使って，乱数生成や量子機械学習を行います．

第3章で解説するプログラムはGoogle Colaboratoryで実行できます．

```
https://colab.research.google.com/?hl=ja
```

世界最速のシミュレータ…Qulacsの紹介

● 量子コンピュータの優位性が得られるかどうかは分かっていない

第1章や第2章で扱った例は1量子ビットや2量子ビットの例だったので，このような簡単な問題をわざわざ量子コンピュータを用いて解くのはなぜかと思うかもしれません．全くその通りです．説明を簡単にするために，少数の量子ビットを用いて説明をしました．

実際には，コスト関数を定義する行列Aやパラメータ付き量子状態$|\psi(\theta)\rangle$は古典コンピュータでシミュレーションが難しくなる数十量子ビットでの実行で量子コンピュータの優位性が得られると期待されています．例えば，30量子ビットとすると量子状態の次元は，$2^{30}\simeq 10^9$になります．50量子ビットであれば，$2^{50}\simeq 10^{15}$次元になり，厳密な対角化や，ブルートフォース（力まかせ）探索などは難しくなります．

そのような状況で，うまくパラメータ付きの量子回路を用いてコスト関数を最小化することができれば，古典コンピュータに対する優位性が得られるかもしれません．実際にはノイズの問題や測定による統計誤差などの問題もあり，このような変分量子アルゴリズムで優位性が得られるようなアプリケーションを見つけることができるかはまだよく分かっていません．

● 量子コンピュータ・シミュレータの必要性

いずれにせよ，本格的に量子アルゴリズムの開発や実装を行うためには，もう少し大規模な系をシステマチックに取り扱う必要があります．もちろん量子コンピュータ実機を直接利用することができればよいのですが，開発途上でありノイズの問題など理想的な結果を得ることが難しい場面もあります．

そのような中で今ますます重要になっているのが，既存のコンピュータを用いた量子コンピュータのシミュレータです．既存のコンピュータの半導体回路設計においても回路シミュレーションや論理シミュレーションなどさまざまなシミュレーションが活用されてきました．ここでは，Pythonインターフェースを持つ世界最速のオープンソース量子コンピュータ・シミュレータQulacsによる量子プログラミングを紹介します．

● Qulacsを用いた20量子ビットの乱数生成シミュレーション

▶ステップ1…Qulacsのインストール

Qulacsは，次のコマンドからインストールできます．

```
pip install qulacs↵
```

▶ステップ2…量子ビットの確保

n量子ビットの状態の確保は，次のように実行できます．

```
from qulacs import QuantumState
nqubits = 20
state = QuantumState(nqubits)
```

この場合，20量子ビットが確保されたので，2^{20}次元の複素ベクトル，つまり約16Mバイトのメモリを使っていることになります．

▶ステップ3…量子ゲートを作用

これらの量子ビットに対して，量子ゲートを作用させることで量子計算をシミュレーションしていきます．第2章で紹介をした量子ゲートを含め，さまざまな量子ゲートがあらかじめ用意されています．または，numpy arrayから量子ゲートを定義することもできます．

例えば，全ての量子ビット（20量子ビット）にアダマール・ゲートを作用させてみましょう．

```
from qulacs.gate import H
for i in range(nqubits):
    H(i).update_quantum_state(state)
```

これで，20量子ビットの2^{20}通りの状態が重ね合わされた状態が作れます．