

# 量子統計物理学的シミュレーションによる量子ビット回路の最適化（継続-9）

## Computer Simulation Study of Quantum Information Processing

伊藤 伸泰 東京大学大学院工学系研究科  
Nobuyasu Ito School of Engineering, The University of Tokyo

フィッシャー・クルト<sup>†</sup> マツチス・ハンス・ゲオルグ<sup>††</sup> 湯川 諭<sup>†††</sup>  
Kurt Fischer<sup>†</sup> Hans-Georg Matuttis<sup>††</sup> Satoshi Yukawa<sup>†††</sup>  
<sup>†</sup>東京大学大学院工学系研究科（現所属：徳山工業専門学校）  
<sup>††</sup>東京大学大学院工学系研究科（現所属：電気通信大学）  
<sup>†††</sup>東京大学大学院工学系研究科（現所属：大阪大学）

<sup>†</sup>School of Engineering, The University of Tokyo (present: Tokuyama College of Technology)

<sup>††</sup>School of Engineering, The University of Tokyo (present: University of Electro-Communications)

<sup>†††</sup>School of Engineering, The University of Tokyo (present: Osaka University)

研究期間 平成 13 年度～平成 17 年度

### 概要

物理現象の量子力学的な描像に基づく量子情報処理・量子情報通信では、古典力学的な描像に基づいたこれまでの IT 技術では不可能あるいは困難と考えられている情報処理・情報通信に途を拓くものと期待されている。一方、統計物理学・物性物理学はじめ計算科学諸分野では、量子力学的対象を理論的に扱い現象を記述・予測するための方法を、古典情報の枠組みの中で開発し応用してきた。量子情報の実現に際しては、こうした古典情報による方法を活用することが有効な場合があると期待される。本研究はこうした可能性を、計算統計物理学の立場から探ることを目的とした。本研究の結果、量子情報処理回路の量子モンテカルロ法によるシミュレーション技術を拓き、また、ヒルベルト空間の全状態の変化を計算するシミュレーションソフトウェアを開発した。さらにこれらを応用し、36qubit までの量子情報処理回路の解析に成功した。

### Abstract

Simulations of quantum information processing circuits with presently available classical computer are challenged. By applying quantum Monte Carlo method with Hubbard-Stratnovich transformation technique, some circuits like quantum Fourier transform are successfully simulated, but it did not work for every circuits. For example, it fails for Simon type period finding algorithm. With direct simulation approach, we succeeded to simulate up to 36 qubit circuit using 1TB memory and TFLOPS parallel computers. CAD for quantum computer named QCAD is also developed.

### 1. まえがき

量子情報技術、特に量子計算回路は実現に向けて現在活発に研究が進められているが、現実のものとなるまであるいは実用後も、その動作を確認し設計するには古典的な計算機によるシミュレーションが不可欠であると考えられる。本研究では、古典計算機による量子計算回路のシミュレーション技術を確認し、より効率的なアルゴリズムを研究することを目標として5年間、研究を行った。

この目標に向けて、量子ビット回路を様々な手法でシミュレートし、収束性や効率について調べた。以下では、まず量子ビットの指数関数的な数にのぼる量子力学的な全状態を追跡する方法とその成果を紹介し、さらに量子統計物理学で培われてきた量子多体系のモンテカルロシミュレーション法の活用と応用とを試みた結果を紹介する。

### 2. 研究内容及び成果

量子情報回路の動作を現在の古典コンピュータでシミュレートする方法としては、大きく分けて2通りある。系の量子状態を計算機上に用意しすべての変化を追跡する方法と、状態をサンプリングして観測量を統計推定する方法とである。前者を「全状態追跡法」、後者を「モンテカルロ法」と、以下では呼ぶことにする。

全状態追跡法の場合、系のヒルベルト空間の基底ベクトルすべての係数を用意し、量子回路の動作に沿って係数に

ユニタリ変換を施してゆくことになる。量子ビット  $n$  個からなる系の場合、ヒルベルト空間は  $2^n$  次元であり、 $2^n$  個の係数を扱うことになる。量子ゲートで用いるユニタリ変換行列は、たかだか数個の量子ビットの相互作用しか持たないため、行列要素の数も  $2^n$  個程度度の素行列となる。このためユニタリ変換に要する(古典的な)計算量も  $2^n$  程度である。よって古典コンピュータによる全状態追跡法では、量子情報処理で期待される処理の抜本的な高速化は得られないが、回路の動作の確認、素子の設計に際しては有効な方法となりうる。

本研究では、オランダ・フローニンゲン大学の Hans De Raedt 博士、ドイツ・ユーリヒ KFA の B. Trieu 博士、名古屋大学の渡辺宙志博士らとともに、さまざまな量子計算回路をさまざまな計算機を使って解析し、要した計算機時間を評価した。詳細は現在投稿中である。

全状態追跡法は、現在の計算機による量子情報処理回路シミュレーションの正攻法ともいべきものであるため、現在あるいは近未来の古典的な計算機によりどの程度の解析が期待できるのかを明らかにすることは量子情報処理技術の生命線ともいえよう。このため本シミュレーションでは現在、最高速級の計算機で広く使われている並列型計算機による解析が重要と考え、以下の計算機などを使った。いずれもテラバイト程度の記憶容量、テラフロップス程度の演算速度をもつ計算機である。

表 1

計算機	IBM Regatta p690+	IBM Blue Gene/L	日立 SR11000/J1
メモリ	5.2TB	3.1TB	2.8TB
CPU	Power 4+	Power PC970	Power5
クロック	1.7GHz	0.7GHz	1.9GHz
CPU 数	1312	12288	2048

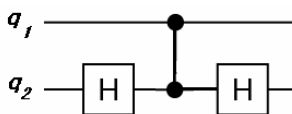
シミュレーションには並列アーキテクチャに最適化したプログラムを用い、1TB の記憶領域をつかって 36 量子ビットまでの回路を扱うことができた。整数の和や量子フーリエ変換の回路を使ってシミュレーションに要した計算時間も計測した。その結果、1TB の記憶容量を使って 12 ビットの整数 3 つを足す 36 量子ビットの量子計算回路を、量子ゲート 1 つあたり 10 秒程度の経過時間でシミュレートすることができた。

このシミュレーション結果から、以下の結論を得た：

- (1)量子計算回路のシミュレーションは、並列あるいはベクトル計算機を用いることにより効率的に実行することができる。
- (2)1TB, 1TFLOPS の計算機を使うことにより、36 個の量子ビットからなる量子計算回路を、平均的には 1 ゲート 10 秒程度の実経過時間でシミュレートすることができる。量子ゲートの入力量子ビットが偏っていれば、さらに短い時間でシミュレートすることができる。
- (3)現在視野に入っている 10 PFLOPS 級の計算機（京速計算機）では、50 個程度の量子ビットからなる量子演算回路をシミュレートできると期待される。
- (4)京速計算機以降の（古典）計算機の成長をいわゆるムーアの法則的と仮定すると、18 ヶ月毎に扱える量子ビットが 1 つずつ増えてゆくと期待される。量子計算機の実用化に当たっては、この成長速度を大きく上回る開発速度が必要である。

全状態追跡法の場合にはヒルベルト空間の次元の大きさが実効上の問題となるが、この問題は量子多体系一般に共通する問題であり、特に熱平衡状態での物性物理学で研究が進んでいる。そして量子多体系を（古典的な）計算機でシミュレートする際には、全状態追跡法と並んでもう一つ、モンテカルロ法が有効である場合が示されている。

量子計算回路の基本ゲートである CNOT ゲートは、



という回路に等価である。すなわち

$$H_2(1 - 2q_1q_2)H_2$$

と表現できる。ここで演算子の添え字は、どちらの量子ビットに作用する演算子であるかを表わす。ここで 2 つの量子ビットの相互作用を表している第 2 項  $(1 - 2q_1q_2)$  は、

$$1 - 2q_1q_2 = \sum_{k=1}^n w_k(a_k + b_kq_1)(c_k + d_kq_2)$$

という形に書き換えることができる（ハバード・ストラトノビッチ変換）。本研究では、この等式を使うことにより量子計算回路のモンテカルロシミュレーションが可能となることを提唱し、実証した。

この方法による量子モンテカルロ法の特徴は、ハバード・ストラトノビッチ変換によって得られた古典変数間には相互作用がない、すなわち素朴なモンテカルロサンプリ

ングがランダムサンプリングとなるという点である。このおかげで、相互作用系にありがちな遅い緩和などの収束阻害要因はない。この結果は論文[1], [2], [3]に公表している。

以上のようなハバード・ストラトノビッチ変換を使った量子モンテカルロ法により、さまざまな量子計算回路のシミュレーションを試みた。その結果、以下の結論を得た (1)本研究で提唱したハバード・ストラトノビッチ変換による量子情報処理回路の量子モンテカルロ法は、正しく収束する場合がある。理想的なシミュレーション下では一般にどのような回路でも収束すると期待されるが、試みた回路の中には効果的な収束が確認できなかった回路もあった（Simon 型の周期計算アルゴリズムの場合である）。

- (2)この方法ではハバード・ストラトノビッチ変数間には相互作用はなく、モンテカルロサンプリングはランダムサンプリングとなる。このため収束を妨げる異常な緩和は現れず、またマルチカノニカル法他による収束加速も期待される場合が多い。(3)どのような分解公式を使えば最善の量子モンテカルロアルゴリズムになるかは、分解に伴う分散などを最小化することにより得られる。

最後に量子計算回路を扱う上で有用な量子計算回路用の CAD プログラム「QCAD」を開発した。このプログラムは、Microsoft Windows 上で動作し、GUI、全状態追跡法によるシミュレーター、各種出力を備えたものである。現在、本研究で開発したプロトタイプを元に、渡辺宙志博士（名古屋大学）らがさらに発展させたものが [http://www.phys.cs.is.nagoya-u.ac.jp/~watanabe/qcad/index\\_j.html](http://www.phys.cs.is.nagoya-u.ac.jp/~watanabe/qcad/index_j.html) でフリーウェアとして公開されている。

### 3. むすび

本研究により、古典計算機による量子情報処理回路シミュレーションについて明確な展望が得られた。今後 10 年間程度で 50 量子ビットからなる量子計算回路のシミュレーションが実現可能となると予想される。さらに今世紀の終わりまでには 100 量子ビット程度が可能となるかもしれない。また量子情報処理回路の動作をシミュレートする量子モンテカルロ法に初めて成功した。この技術の今後の発展いかんでは、ある種の量子回路のシミュレーションは著しい進歩を遂げる可能性がある。さらに量子計算回路用の CAD「QCAD」を開発し、公開した。

以上の研究成果は、今後の量子情報処理技術をソフト面から支える基礎技術となると確信している。

#### 【誌上发表リスト】

- [1] H. G. Matuttis, K. Fischer, N. Ito and M. Ishikawa, "Auxiliary Field Methods for the Simulation of Quantum Computation Circuits", Intern. J. Mod. Phys. C Vol. 13 No. 7p.917—929 (2002)
- [2] K. Fischer, H. G. Matuttis, N. Ito and M. Ishikawa, "Quantum-Statistical Simulations for Quantum Circuits", Intern. J. Mod. Phys. C Vol. 13 No. 7 p.931—945 (2002)
- [3] K. Fischer, H.-G. Matuttis, S. Yukawa and N. Ito, "Quantum Computing Simulation using the Auxiliary Field Decomposition", in "Computer Simulation Studies in Condensed Matter Physics XVI" ed. D. P. Landau, S. P. Lewis and H. B. Schüttler, p.79-85 (Springer-Verlag, 2004)