

古典力学から量子力学へ そして量子計算へ

編著 一色 浩

測位技術振興会

BROCADE CHAIN – TRUST BOOK

法律により本ファイルの
複写・コピーは
禁止されています。

BROCADE CHAIN – TRUST BOOK

発行：SAPT 出版局

Mercari Code: Axion

Serial No. 24-2

本書は、教育機関である測位技術振興会の知能・情報・エネルギー分科会における教材として編纂されたものである。
研究・教育を目的とする引用については、著作権法第 32 条
1 項に基づいて実施されるものとする。

本書は、学術同人誌として当該規定に則るものとする。

古典力学から量子力学へ そして量子計算へ

編著 一色 浩

まえがき

ここ数年来、量子計算機に対する期待が大いに高まっている。最新型のスーパーコンピューターでも1万年掛かる計算が僅か3分弱で計算できるとのことで、実に未曾有のことである。社会に大変革をもたらすことは間違いないであろう。

つい最近にデープラーニングに代表される人工知能（AI）による大変革が取り沙汰されたばかりである。最近、対話型 AI というものが登場して、人類社会に大きな革新を起こすことが予想されているが、その危険性も危惧されている。量子計算機による大変革はこれとは全く異質のものであろう。AI は凡人を非凡人にすることを可能にすると思うが、量子計算機のもたらす変革は、我々の想像をはるかに超えるものになろう。

古典計算機（本書では在来型の電子計算機をこのように呼ぶ）が生まれて100年足らずであるが、古典計算機は驚くべき進化を遂げた。計算速度の向上が一貫して追求されてきた。しかし、古典計算機では計算できない問題が数多くあり、さらなる速度の向上が求められている。そのような状況の下に登場したのがスーパーコンピューター（スパコン）である。スパコンの原理は、並列計算である。しかし、並列計算の大規模化をどんなに進めても現代文明の要求には不十分なことが明らかになって来た。

そんな中で、今まさに登場しつつあるのが量子計算機である。量子計算機とは何かを一言で言えば、「量子力学の原理に基づく超並列計算機」であると言えよう。量子力学の原理とは何かと言うと、「重ね合わせの原理」である。量子は「いろんな状態が重なり合った状態で存在できる」ということである。要するに「超並列状態」を取り得るということである。

このような量子計算の可能性を最初に指摘したのは、物理学者のリチャード・ファインマンである。1981年の「物理学と計算」会議の基調講演において次のような指摘を行った。すなわち、「量子の振る舞いをシミュレーションするには通常のコンピューターでは限界があり、量子力学の原理に従うコンピューターが必要だ」と述べた。量子力学の原理を使えば、究極の並列計算機が実現できると解釈できよう。実に驚くべき指摘である。

以上のような観点からみれば、今まさに登場しようとしている量子計算機を、いち早く理解することが求められているのではなかろうか？量子計算機の基礎となっている量子力学そのものは、誕生以来すでに100年を超えるばかりでなく、半導体を始めとする様々な製品を生み出す原動力となって来た。

しかし、そのような恩恵にも関わらず量子力学や量子計算機に対する我々の理解は、ごく少数の専門家を別として、極めて浅いと言わざるを得ない。量子力学は世界の根源を支配する物理法則であるので、このような状況は奇妙なことである。何故だろうか？

これには二つの理由があるように思われる。一つは我々が五感を通して直接経験できる物理現象の大半は、ニュートンが創始したいわゆる古典力学で説明できる。野球やサッカーのボールの運動はそのような例である。その著しい特徴はボールの位置と速度は運動方程式により一意的に決定できることである。古典力学によれば、天体の運動は未来永劫に渡って完全に予測できる。また、物体と物体との相互影響は必ず近接作用として行われ、地球上の物体が月の上の物体に直接影響を及ぼすことはない。すなわち、遠隔作用はない。

もう一つの理由は、量子計算機の根幹である量子現象である。一番よく知られているのは、粒子の量子性と2重性であろう。量子性は質量やエネルギーのような物理量は、いくらでも小さくはできず、最小単位があるということである。また、粒子は波動であり粒子であるというのが2重性である。ここまでは許容できても、物質の状態は瞬間的に確定できず、複数の状態が重なりあっており、確定できるの

は物質が取る様々な状態の確率で、観測すれば一つの状態になるが、その状態は確率に従う。この確率の意味は、いわゆる不規則現象とは異なる。不規則現象の場合の確率は、物質の置かれた状況を完全に把握できないことを近似的に扱う手段である。しかし、上述の量子力学の確率は、近似にらず自然の本質である。

では、何故古典力学と異なる量子力学のこのような特徴が、我々の理解を妨げるのであろうか？それは、要するに五感に合わないということであろう。五感は大脳の働きの結果形成されるものであろうから、古典力学に占領されてしまった大脳が異物として排除するためであろう。これは、外国語の習得が難しいのと似ていると思う。我々は、誕生直後から母国語のシャワーに四六時中さらされる。それが妨げになって、外国語の習得には大変時間が掛かってしまう。母国語の知識は重要なものと思うが、それに囚われないことである。我々の思考、特に深い思考は母国語に頼らざるを得ないので、母国語を支配者とせざるを得ないが、外国語を母国語の囚人にしてはならない。

全く同じ理由で、古典力学に慣れ親しんで来た者にとっては、新たに量子力学を学ぶからと言って古典力学を軽視してはいけないであろう。これはどちらが正しいということではないので、古典力学の理解を深める機会とすべきように思う。

最後に、シュレィンガーの波動方程式というように、古典的な波動現象や振動現象の物理や数理に馴染んでいる者にとっては、量子力学や量子計算機は実に分かり易い。ただし、なかなか受け入れ難い。また、線形の現象であるので、線形代数の知識があることが望ましい。確かに奇妙な世界であるが、極めて楽しめる世界であることに間違いない。本書の目的とするところは、著者と同様に古典力学に馴染んで来た者が、新たに量子計算機を勉強しようとする際の助けになればと思う。そのための労は惜しまなかった積りである。

しかし、量子現象を理解するためには、一つだけ必要なことがある。いわば、古典力学という母国から量子力学という外国を訪れる旅人になる訳であるので、当然のことながら、言葉も習慣も異なる。「郷に入れば郷に従え」と言う言葉があるように、外国の異文化の価値を認めて溶け込む努力を惜しまぬという誓いを立てて、異文化を受け入れ、さらに楽しむことである。

要するに、「神」を信じなければ宗教を信じることはできない。これと同じことで、「重ね合わせ」を受け入れなければ量子力学を受け入れることはできないであろう。「神」とか「重ね合わせ」は概念としては存在可能である。「重ね合わせ」が物理的実在であると強要することはできないが、現に我々は日常生活でその恩恵を受けていることは否定できないであろう。

第6章までは量子計算の数理的な側面について述べる。量子計算機がどのようなハードウェアで構成されるかについてはほとんど説明されていない。これでは真の意味で量子計算を理解するには不十分であろう。そこで、第7章にハード面の解説を述べている。

一方、量子計算機のハードウェア開発はかなり進んだと考えられるが、まだまだ多くの技術的課題が山積している。本格的な量子計算機の登場はまだかなり先になろう。このような背景の中で、量子計算に刺激されて在来型の計算機を革新する技術が生まれつつある。いわば疑似量子計算機とも言うべきものが登場しており、組み合わせ爆発問題に大きな成果を挙げつつある。このような疑似量子計算機の解説を第8章で行う。

なお、本書は測位技術振興会（SAPT）の「測位・情報・エネルギー分科会」が2022年6月から2023年8月に行った調査研究の成果報告である。本委員会の委員を下記に記す：

分科会長 一色 浩 （有）数理解析研究所（第1から第6章、ノートA、B執筆）

企画助言 浅里幸起 一般財団法人 宇宙システム開発利用推進機構
委 員 毛利篤史 三菱電機(株) (第7章執筆)
委 員 板東幹雄 (株)日立製作所 (第8章執筆)
委 員 河村政博 RIO NET ASIA 代表
委 員 杉本末雄 立命館大学名誉教授
委 員 影本 浩 東京大学名誉教授
委 員 宮武克昌 測位技術振興会理事

分科会委員はいづれも古典力学の分野で仕事をして来たので、量子力学の学習には種々の困難があったが、活発な議論を通して一応の習得ができたものと理解している。また、古典力学分野のものが量子力学さらに量子計算を学習する上での経験が、我々と同様の挑戦を試みる方々に何らかのお役に立つのではと思い、本書を世に問うことにした。誤った解釈をしないように細心の注意と努力を払ったつもりであるが、誤った解釈が紛れ込んでいるかも知れない。そのような場合には御寛恕の程をお願いしたい。

分科会のディスカッションに使った PPT ファイルの内容も学習に役立つと思うので、本書の後半に含めてある。本文と併せて利用して頂きたい。

酷暑のなか
一色 浩
2023. 08. 31

目 次

第 1 章 量子計算機とは	1
第 2 章 量子力学とは	2
2.1 量子力学と量子計算の歴史	2
2.2 量子の基本法則	3
2.3 シュレディンガーの波動方程式	5
2.4 ボルンによる波動方程式の確率解釈	7
2.5 演算子のエルミート性	8
2.6 波束の不確定性	9
2.7 波束の時間進化	10
2.8 フーリエ変換の復習と解釈	11
2.9 演算子の期待値	12
2.10 演算子の時間進化	12
2.11 オブザーバブル（観測可能な物理量）とハミルトニアン	13
2.12 エルミート演算子の固有値：スペクトル理論	13
2.13 量子力学から古典力学へ（エーレンフェストの定理）	14
2.14 モードに分解されたシュレディンガーの方程式	15
2.15 2 準位原子の問題	16
2.16 ボーアとアインシュタインの論争	18
第 2 章の参考文献	19
第 3 章 量子計算の理論	20
3.1 量子計算の特徴	20
3.2 見ることができる量子的現象 1…ヤングの 2 重スリット	23
3.3 見ることができる量子的現象 2…マッハ・ツェンダー干渉計	26
3.4 重ね合わせと絡み合い	31
3.5 量子計算	32
第 3 章の参考文献	33
第 4 章 モード展開されたシュレディンガー方程式	34
4.1 古典力学における弦の運動	34
4.1.1 非加算無限自由度（無限長，分布質量）の弦の波動	34
4.1.2 加算無限自由度（有限長，分布質量）の弦の振動	35
4.1.3 有限自由度（有限長，離散質量）の弦の振動	36
4.2 量子力学における 1 量子の運動	37
4.2.1 非加算無限自由度（無限領域，連続確率密度分布）の 1 量子の運動	37
4.2.2 加算無限自由度（有限領域，無限離散確率）の 1 量子の運動	38
4.2.3 有限自由度（有限領域，有限離散確率）の 1 量子の運動	39

第4章の参考文献	40
第5章 量子アニーリング問題	41
5.1 量子アニーリングの現状	41
5.2 量子アニーリングの理論	41
5.2.1 量子アニーリングによる大域最小値探索	41
5.2.2 量子アニーリングの理論	42
5.3 量子アニーリングの計算例	46
5.3.1 一変数の場合の量子アニーリング大域最小値探索	46
5.3.2 多変数の場合の量子アニーリング計算	52
5.3.3 量子アニーリング計算の特長	53
5.4 拘束条件のある場合の最小値問題	55
5.4.1 等式拘束条件のある場合	55
5.4.2 不等式拘束条件のある場合	55
5.5 量子アニーリングによるニューラル・ネットワークの学習	57
5.6 アニーリングによる連立1次方程式の解法	60
5.6.1 最小値問題に変換	60
5.6.2 10進変数を2進変数に変換	61
5.7 まとめ	62
第5章の参考文献	63
付録A C言語で書かれた量子アニーリングの計算コード	63
第6章 量子ゲート問題	86
6.1 量子計算の大前提	86
6.2 量子論理ゲート	87
6.2.1 古典論理ゲート	87
6.2.2 量子論理ゲート	87
6.2.3 いろいろな量子論理ゲート	88
6.2.4 絡まった状態	91
6.3 量子テレポーテーション	92
6.4 ドイッチージョサ(Deutsch-Jozsa)のアルゴリズム	94
6.4.1 ドイッチのアルゴリズム	94
6.4.2 ドイッチージョサのアルゴリズム	95
6.5 ショアのアルゴリズム	98
6.5.1 公開鍵暗号	98
6.5.2 量子計算によるショアのアルゴリズムを導くための準備(素因数分解)	99
6.5.3 量子離散的フーリエ変換	101
6.5.4 位相推定問題	108
6.5.5 位数計算	111
6.5.6 ショアのアルゴリズムの手順	116

6.6 あとがき	119
第6章の参考文献	120
第7章 量子計算機実現の方式概要	121
7.1 量子計算機 5 方式概要	122
7.2 超電導方式	123
7.3 光量子方式	124
7.4 イオントラップ方式	125
7.5 シリコン方式	126
7.6 冷却原子方式	127
7.7 量子計算機の実現方式別の将来展望	128
第7章の参考文献	131
第8章 実用化された疑似量子コンピューティング	132
8.1 アニーリングマシン・イジングマシン概説 [1]–[3]	132
8.2 商用化済みの疑似量子アニーリングマシン [3]	133
8.3 NTT 社 Coherent Ising Machine [5]–[6]	134
8.4 東芝社 Simulated Bifurcation [7]	135
8.5 NEC 社 Vector Annealing [8]–[9]	136
8.6 富士通社 Digital Annealer [10]	137
8.7 日立社 CMOS Annealing [11]–[14]	138
8.8 FIXSTARS Amplify Annealing Engine [15]	139
8.9 終わりに	140
第8章の参考および参照 Web サイト	140
ノート A 複素ベクトル空間	142
ノート B 量子ゲート問題の基礎	146
プレゼンテーション資料	
古典力学から量子力学へそして量子計算へ	P-1
第2章 量子力学とは	P-1
第3章 量子計算の理論	P-27
第4章 モード展開されたシュレディンガー方程式	P-42
第5章 量子アニーリング問題	P-54
第6章 量子ゲート問題	P-79
第7章 量子計算機の方式調査	P-115
第8章 疑似量子アニーリングに関する調査(イジングマシン)	P-119

第1章 量子計算機とは

量子計算というアイデアは思いのほか新しいもので、1981年にアルゴンヌ国立研究所のポール・ベニオフが古典計算機（現在の計算機）で可能なことは量子システムでも可能と考えたこととされる。

しかし、もっと重要と思われるのは、ノーベル賞受賞の物理学者であるリチャード・ファインマンによる1982年の次のような指摘であろう。「古典計算機で自然の根幹である量子現象を短時間でシミュレートすることはできない。指数関数的な時間が掛かってしまう。量子力学の原理に立脚する新しい計算機に依らねばならない」、これは実に重い言葉である。正に、量子計算機の本質を言い尽していると思う。

1985年にオックスフォード大学のデイビッド・ドイチは、量子計算の数理的基礎を確立した。その中でとりわけ重要なのは量子並列の概念であった。さらに、量子並列の概念を具現する量子回路（現在ドイチ問題と呼ばれるもの）により、古典的な方法では考えられない量子並立（量子的重ね合わせ）の原理を示した。

10年ほど画期的な進展はなかったが、1994年AT&Tベル研究所のピーター・ショアが大きな整数の素因数分解を短時間で行う画期的な量子アルゴリズムを発見した。現在広範に使われているRSA暗号（公開鍵暗号）は、「大きな整数の素因数分解」が短時間でできないことに基づいている。従って、これが短時間でできるようになったら、一大事である。もっとも、その後量子暗号というさらに強固な暗号システムができていたので、そのような危険は回避されそうである。

1995年に、デイビッド・ドイチは2種類の基本的な量子ゲートがあれば、どのような量子回路でも実現できることを示した。

1996年に、グローバーは、データベースの高速検索アルゴリズムを発見した。

量子計算は大別すると、量子ゲート問題と量子アニーリング問題になる。前者は汎用計算機を目指すもので、100万量子ビットが目安であるが、2013年の時点で、実現までに今後10年ほど掛かると見られている。後者は組み合わせ爆発の最小値問題（最適問題）をアニーリング（焼き鈍し）で解かんとするものである。

2011年に、カナダのD-Waveシステムズ社は、世界初の商用量子コンピューター D-Wave One を公開した。上述の量子アニーリングに関するものである。これは組み合わせ爆発問題の一種であるイジング問題を解く専用機である。イジング問題とは、結晶格子上の磁気スピン分布を論じる問題を起源とする2値変数を変数とする2次関数の最小値問題である。東京工業大学教授であった西森秀稔と大学院学生であった門脇正史が開発した量子アニーリング理論を基にしている。

実は、量子ゲート問題と量子アニーリング問題は双方に変換可能で、前者でできることは後者で可能で、後者でできることは前者で可能であるとされる。特に、量子アニーリングはD-Wave社が専用マシンを開発しており、大きな関心を集めている。

古典計算機では1万年掛かるような問題が、D-Waveの計算機では2分で解けることが示されたので、大反響を巻き起こし、現在に続く量子計算機の大ブームを招来した。しかし、その限界も見え始めている。本来、その能力がもっと順調に伸びても良いと思えるが、そのようにならない理由の一つは、実は量子計算機は一つの大きな技術的課題を解決していないからである。

その課題とは、コヒーレンスと呼ばれる量子状態を維持できる時間が短いためであろう。そこで出て来たのが古典計算機を使って純正量子計算機と似た計算をさせる疑似量子計算機である。当然のことな

がら純正量子計算機の性能には及ばないが、現状では安定性と信頼性に優れ、小型化も可能とされている。

第2章 量子力学とは

物理学者であるリチャード・ファインマンの言葉に、「相対性理論は誰でも理解できるが、量子力学が分かっているというやつは嘘つきだ」という言葉があるそうである。また、量子力学の創始者の一人であるニールス・ボーアは「もし初めて量子力学を学んだ時に何の疑問も抱かないのなら、それは量子力学について何も理解していない」と言ったそうである。

量子力学は、人が直接見ることのできない極微小な原子の世界を支配している物理法則である。我々が五感で体感できる世界の常識では理解できない奇妙な世界であるが、量子力学を当てはめて得られる予測は、実験結果を完全に説明して外れることがない。そのような観点から量子力学は「正しい」とされている。現在では、古典力学も包含する根源的な物理とされている。

量子力学は我々の日常経験と合わない奇妙な理論であるが、古典力学よりも難しいものではない。それは線形理論だからである。柔軟に受け入れれば、とても魅力的な理論である。

2.1 量子力学と量子計算の歴史

量子力学と量子計算の歴史を概観してみたい。厳密に論ずるのは科学史の専門家に任せるとして、個人的には以下のように捉えている：

- (1) プランクとアインシュタインによる量子の発見
- (2) アインシュタインとド・ブロイによる粒子の2重性
- (3) シュレディンガーとハイゼンベルグによる粒子運動の記述
- (4) ボルン、ボーアらによる確率波の概念の導入
- (5) ベニオフ、ファインマン、ドイチ、ショアラによる量子計算の基礎概念、数理、アルゴリズムの基礎構築

詳しい説明は本書の目的から外れるので、ざっと説明したい。(1)の量子の発見は、黒体放射のスペクトルの研究において、ドイツの物理学者マックス・プランクが1900年に見出したものである。プランクは全波長領域で適用できる黒体による熱放射のスペクトルの温度特性を確立した。いわゆるプランクの公式である。

プランクはこの法則の導出において、物体が光を吸収または放射する時、そのエネルギーは、エネルギー素量（現在ではエネルギー量子と呼ばれている） $\varepsilon = h\nu$ の整数倍でなければならないと仮定した。物質を構成する分子の熱振動のエネルギーの離散性を仮定して導かれた。プランク自身はこの仮定がさほど重要と思っていなかったようであるが、この量子仮説は正しく量子力学を産み出す跳躍板となった。アルバート・アインシュタインの光量子仮説に直結している。

(1)と(2)にまたがる1905年のアインシュタインの光量子仮説は、つぎのような現象（光電効果）を見事に説明した。その現象とは、「物質に光をあてると電子が飛び出すが、電子の放出は、ある一定以上大きな振動数の光でなければ起こらず、それ以下の振動数の光をいくら当てても電子は飛び出してこない」というものである。

光量子仮説では、光は波動であるが光電効果を説明するためには、粒子と考えた方がすっきり説明できるとする。波動の元々のイメージは連続的なもので、光の強さは波動の振幅であるから、どんな周波

数の光でも、強さを強くすれば電子が飛び出すことになると思うのが自然である。しかし、光量子仮説では光を粒子（現在では光子と呼ばれる）と見做し、粒子一つ一つが $\varepsilon = h\nu$ を持つとし、光の強さは粒子の数に比例するとする。電子1個に1度に衝突するのは光子1個とすれば、光子1個のエネルギーが小さな場合は、いくら光を強くしても（光子の数を増やしても）電子が飛び出すのに必要なエネルギーは与えられない。だから、電子は飛び出さない。

1905年にアインシュタインは、科学ジャーナル・アナーレン デア フィジークに画期的な論文4篇を発表した。すなわち、光電効果、ブラウン運動、特殊相対性理論、 $E = mc^2$ （質量とエネルギーの等価性）に関するものである。この4篇の論文は、原子の世界、空間、時間、質量、エネルギーといった基本的な概念に対する科学的認識に根源的な変革を起こし、現代物理学の基礎を築いたと言える重要なものである。1921年に光電効果に関してノーベル賞を受賞した。このような画期的な論文が1年で発表されたことから、1905年は奇跡の年と呼ばれている。この当時、アインシュタインは、スイスの特許庁で審査官をしていた。これは大変興味深いことで、学者の業績と置かれた環境に関する示唆に富むことである。

(2)の量子の2重性に関しては、アインシュタインは光子は波動であるとともに粒子であると主張した。これに対して、ド・ブロイは電子は粒子であるとともに波動であると主張し、波動の群速度は電子の運動速度 v であり、波長 λ は h/p となることを示した。ここで、 h はプランクの定数、 p は電子の運動量である。ド・ブロイはこれでノーベル賞を受賞した。この波は電子以外にも陽子や中性子にもみられる。理論的には運動する物体すべてにみられる。突拍子もない考えのようではあるが、この理論が量子力学へとつながっていった。電子顕微鏡が極めて高倍率であるのは、電子波の波長は光波よりもずっと短くできるからで、ド・ブロイのアイデアを具現したものであろう。

(3)の粒子の運動の記述であるがエルヴィン・シュレディンガーによるものとヴェルナー・ハイゼンベルグによるものの2種類あるが、ニュートンの古典的な運動方程式とは大いに異なる。古典的な運動方程式は、運動する粒子の位置と速度（運動量と考えても良い）の時間発展を述べるものであるが、シュレディンガー方程式は波動関数の時間発展を述べるものである。

この方程式が導かれたとき、シュレディンガーが考えたのは、想定された電子の位置や速度を直接対象にしたものではなくて、電子の速度で移動する波動を漠然と想像し、この波動の満たすべき方程式を求めることであった。この波動と電子の運動とが、具体的にどんな関係になるのかは明示しなかった。波動が何を表すかを人に問われたときに、「知らない」と答えたそうである。

そこで登場するのが(4)のマックス・ボルンである。彼は波動関数の絶対値の2乗が運動する電子の位置の確率密度分布を与えると考えた。いわゆる確率解釈である。現在、この確率解釈が矛盾がなくて最も正統的な解釈とされている。古典力学では粒子の位置や速度は決定論的であったが、確率解釈により量子の世界では決定論的ではなくて、確率論的であるという大変革がもたらされた。

それまで、古典力学に馴染んできたほとんどの物理学者にとっては正に驚天動地のことで、長きにわたって論争が続いた。「神はサイコロを振らない」という有名なアインシュタインの言葉に示されるように、アインシュタインは確率解釈反対者の筆頭であった。シュレディンガーも反対者であった。アインシュタインの解釈では、確率解釈が生まれるのは現在の量子力学には欠けているものがある。未知の変数があるのではというのが、アインシュタインの主張であったが、この考えは否定されている。このような反対論の中で、ニールス・ボーアは首尾一貫して確率解釈を擁護し続けた。長年にわたり、ボーアとアインシュタインの間で交わされた論争は広く知られている。

(5) で量子計算が生まれた背景に触れている。すでに、ベニオフ、ファインマン、ドイツ、ショアらの業績については、第1章で詳しく説明しているので、ここでは省略する。

2.2 量子の基本法則

量子に関する最も基本的な法則および考察は、アインシュタインの光子に関する法則とド・ブロイによる物質波への拡張と思われる。

(a) 光のエネルギーと運動量

アインシュタインは光量子仮説により、光子のエネルギー E と振動数 ν の関係：

$$E = h\nu \quad \text{or} \quad E = \hbar\omega \quad (2.1)$$

および運動量 p と波数 k に関する重要な法則：

$$p = mc = \frac{mc^2}{c} = \frac{E}{c} = \frac{h\nu}{c} = \frac{h}{\lambda} \quad \text{or} \quad P = \hbar k \quad (2.2)$$

を導いた。ここで、 ω と λ は円周波数と波長、 m は光子の質量、 c は光速、 $h = 2\pi\hbar$ はプランクの変数である。

式(2.2)については、少し説明が必要であろう。光子は質量0なので $p = mc$ とすることには抵抗を感じるが、光子の代わりに電子を想定すれば矛盾しない。

(b) ド・ブロイの物質波

アインシュタインは波動である光に粒子性があると考えた。ド・ブロイはこれとは逆に粒子に波動性があると考えた。光子の運動量は $p = h/\lambda$ であるが、あらゆる粒子においてこの式が成り立つと考えた。量子力学誕生以来100年経っているので、現在の我々にとっては当たり前のことのように思えるが、この量子の2重性は、量子ならではの性質で、量子力学の発展に特有の位置を占めている。

上式を変形すると $\lambda = h/p$ であり、 $p = mv$ を代入すると $\lambda = h/p = h/(mv)$ である。この式を用いれば、運動する粒子の波長を計算できる。このときの波を物質波あるいはド・ブロイ波という。特に電子の場合、電子波という。

この波は電子以外にも陽子や中性子にもみられる。理論的には運動する物体すべてにみられる。突拍子もない考えのようではあるが、この理論が量子力学へとつながっていった。

ド・ブロイの速度は位相速度でなくて波束（粒子）の移動速度である。ここで、波束と言うのは、粒子を波のイメージで捉えると波のかたまりが連想されると思うが、このかたまりのことである。式(2.2)で与えられる粒子の運動量と式(2.1)で与えられるエネルギーから

$$p = \hbar k, \quad E = \hbar\omega \rightarrow k = \frac{p}{\hbar}, \quad \omega = \frac{E}{\hbar} \quad (2.3)$$

を得る。

波の速度を計算してみる。速度は、よく知られている方法で求められる。そのために、下記のような1次元波動を考える：

$$\text{mother wave} = Ae^{i(kx - \omega t)} \quad (2.4)$$

この式より、波の位相速度 v_{phase} は

$$v_{\text{phase}} = \frac{\omega}{k} = \frac{E}{P} = \frac{\frac{1}{2}mv^2}{mv} = \frac{1}{2}v \quad (2.5)$$

となり、粒子の運動速度 v の半分である。

一方、波の群速度 v_{group} （波束の移動速度：エネルギーの伝搬速度）は波の分散方程式から求められる：

$$v_{group} = \frac{d\omega(k)}{dk} = \frac{dE}{dP} = \frac{d}{dP} \left(\frac{P^2}{2m} \right) = \frac{P}{m} = v \quad (2.6)$$

従って、粒子の運動速度は群速度に等しいことが確認できる。

(c) 波束による自由粒子の表現

式(2.4)の波は単一波長の波であるが、一般的にはいろんな波長の波が集まったものである：

$$\text{wave} = \int_{-\infty}^{\infty} \Phi(k) e^{i(kx - \omega(k)t)} dk \quad (2.7)$$

波数分布が広帯域で $\Phi(k) = \Phi_0 = \text{const}$ の場合には

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} dk = \delta(x) \quad (2.8)$$

を用いると、波の位相速度が一定とすれば

$$\int_{-\infty}^{\infty} \Phi e^{i(kx - \omega(k)t)} dk = \Phi_0 \int_{-\infty}^{\infty} e^{i(kx - \omega(k)t)} dk = \sqrt{2\pi} \Phi_0 \delta(x - ct) \quad (2.9)$$

となり、位相速度 $c = \omega(k)/k = \text{const}$ で x 軸上を移動する x 軸上の一点に集中する波を表す。ここで、 $\delta(x)$ ディラックのデルタ関数である。

波数分布が全波数に及ばない狭帯域の場合には、波形は図 2.1 のような、いわゆる波束と呼ばれる形状になる。

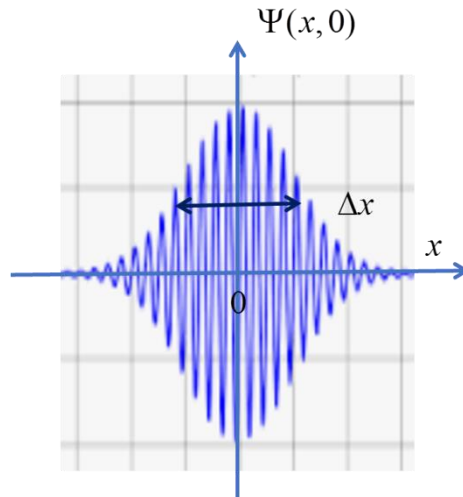


図 2.1 波束

2.3 シュレディンガーの波動方程式

(a) 古典的な波動方程式の導出

まず、古典的な音の伝搬を表す 1 次元の波動方程式について考えて見よう。音の圧力を $p(x, t)$ 、伝搬速度を c とすると音の伝搬方程式は次式で与えられる：

$$\left(\frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} \right) p = 0 \quad (2.10)$$

偏微分方程式(2.10)の解 $p(x, t)$ を

$$p(x, t) = P e^{i(kx - \omega t)} \quad (2.11)$$

と仮定して、式(2.10)に代入すると

$$\left(\frac{\omega^2}{c^2} - k^2 \right) p = 0 \quad \text{or} \quad \frac{\omega^2}{c^2} - k^2 = 0 \quad (2.12)$$

を得る。ここで得られた円周波数 ω と波数 k の関係を分散関係式または分散方程式と呼ぶが

$$\frac{\partial p}{\partial t} = -i\omega p, \quad \frac{\partial p}{\partial x} = ikp \quad (2.13)$$

であるので、 ω と k および $\partial/\partial t$ と $\partial/\partial x$ の間には、以下の関係がある：

$$\left(\frac{\partial}{\partial t}, \frac{\partial}{\partial x}\right) \sim (-i\omega, ik) \quad \text{or} \quad (\omega, k) \sim \left(i\frac{\partial}{\partial t}, -i\frac{\partial}{\partial x}\right) \quad (2.14)$$

この関係を用いると、分散方程式から偏微分方程式を導くことができる。すなわち

$$\left(\frac{\omega^2}{c^2} - k^2\right)p = 0 \rightarrow \left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2}\right)p = 0 \quad (2.15)$$

である。

(b) シュレディンガー方程式の導出

量子力学の基本法則を与える式(2.1)と式(2.2)より、分散方程式を導くことができる。質量 m の速度 v で運動する粒子例えば電子を想定しよう。このとき、粒子の運動エネルギー E は

$$E = \frac{p^2}{2m} \quad (2.16)$$

である。この式に、式(2.1)と式(2.2)を代入すると、分散方程式：

$$\hbar\omega = \frac{k^2\hbar^2}{2m} \quad (2.17)$$

が求まる。

式(2.11)で与えられるような1次元の波動を想定して、式(2.14)を用いれば、容易に偏微分方程式、すなわちシュレディンガー方程式：

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} \quad (2.18)$$

が求まる。ここで、 $p(x,t)$ の代わりに $\Psi(x,t)$ としているが、量子力学では $\Psi(x,t)$ を波動関数と呼ぶ。

式(2.18)は電子のような1個の自由運動粒子のシュレディンガー方程式であるが、ポテンシャルエネルギー $V(x)$ がある場合には、式(2.16)の代わりに、全エネルギー E (E を運動エネルギーに限定して、全エネルギーを H としても良い) は

$$E = \frac{p^2}{2m} + V(x) \quad (2.19)$$

であるので、分散方程式は

$$\hbar\omega = \frac{k^2\hbar^2}{2m} + V \quad (2.20)$$

となる。従って、この場合のシュレディンガー方程式は

$$i\hbar\frac{\partial\Psi}{\partial t} = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right)\Psi \quad (2.21)$$

となる。

さらに、シュレディンガー方程式は

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) \quad (2.22)$$

で定義される演算子 \hat{H} を導入して

$$i\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi \quad (2.23)$$

と書かれる。演算子 \hat{H} はハミルトニアンと呼ばれる。運動量演算子 \hat{p} ：

$$\hat{p} = -i\hbar\frac{\partial}{\partial x} \quad (2.24)$$

を導入して、演算子 \hat{H} は次式のようにも書かれる：

$$\hat{H} = \frac{\hat{p}^2}{2m} + V(x) \quad (2.25)$$

\hat{p} が運動量演算子と呼ばれるのは、式 (2.2) と式 (2.14) より

$$p = k\hbar \sim -i\hbar \frac{\partial}{\partial x} \quad (2.26)$$

が言えるからである。以下の (c) において、演算子について補足する。

3次元の場合には

$$\hat{H} = -\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + V(x, y, z) \quad (2.27)$$

となる。

(c) 演算子とシュレディンガー方程式

式 (2.24) で運動量演算子が登場したが、演算子と固有値は量子力学において重要な概念であるので若干の説明を付け加える。

式 (2.4) で与えられる波動関数を想定するとともに式 (2.2) と式 (2.14) を用いると

$$p\Psi(x, t) = \hbar k\Psi(x, t) = \frac{\hbar}{i} \frac{\partial}{\partial x} \Psi(x, t) \quad (2.28)$$

を得る。式 (2.24) で定義された演算子 \hat{p} を使って式 (2.28) を書き直すと

$$\hat{p}\Psi(x, t) = p\Psi(x, t) \quad (2.29)$$

となる。時間変数 t は $\Psi(x, t)$ にしか含まれていないので無視してよい。式 (2.29) は演算子 \hat{p} の固有値問題を定義しており、この固有値問題の固有値が運動量 p に他ならない。演算子 \hat{p} を運動量演算子と呼ぶ所以である。

同様なことをエネルギー E について考えて見よう。式 (2.1) と式 (2.14) を用いると

$$E\Psi(x, t) = \hbar\omega\Psi(x, t) = i\hbar \frac{\partial}{\partial t} \Psi(x, t) \quad (2.30)$$

を得る。ここで、エネルギー演算子 \hat{E} を

$$\hat{E} = i\hbar \frac{\partial}{\partial t} \quad (2.31)$$

と定義すると、式 (2.30) は

$$\hat{E}\Psi(x, t) = E\Psi(x, t) \quad (2.32)$$

と書き直される。空間変数 x は $\Psi(x, t)$ にしか含まれていないので無視してよい。式 (2.32) は演算子 \hat{E} の固有値問題を定義しており、この固有値問題の固有値がエネルギー E に他ならないことを示している

式 (2.32) をさらに書き直すと

$$\hat{E}\Psi = E\Psi = \frac{p^2}{2m} \Psi = \frac{p}{2m} p\Psi = \frac{1}{2m} p \left(\frac{\hbar}{i} \frac{\partial}{\partial x} \Psi \right) = \frac{1}{2m} \frac{\hbar}{i} \frac{\partial}{\partial x} (p\Psi) = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \Psi \quad (2.33)$$

を得る。これはシュレディンガーの方程式 (2.18) に他ならない。式 (2.33) をさらに書き直すと

$$\hat{E}\Psi = \hat{p}^2 \Psi \quad (2.34)$$

とも書ける。

最後に演算子は

$$\hat{A}\hat{B}\phi = \hat{A}(\hat{B}\phi), \quad (\hat{A} + \hat{B})\phi = \hat{A}\phi + \hat{B}\phi \quad (2.35)$$

という性質があるが

$$\hat{x}\hat{P}\phi - \hat{P}\hat{x}\phi = \hat{x}(\hat{P}\phi) - \hat{P}(\hat{x}\phi) = \hat{x}\left(\frac{\hbar}{i}\frac{\partial}{\partial x}\phi\right) - \frac{\hbar}{i}\frac{\partial}{\partial x}(x\phi) = -\frac{\hbar}{i}\phi = i\hbar\phi \rightarrow \hat{x}\hat{P} - \hat{P}\hat{x} = i\hbar \quad (2.36)$$

であるので、可換ではない。ここで、 x はスカラーなので、 $\hat{x} = x$ としている。

2.4 ボルンによる波動方程式の確率解釈

シュレディンガーの波動方程式 (2.21) は、波動関数 Ψ の物理的な意味付けなしで、式 (2.1) と式 (2.2) を前提にして導かれたものである。シュレディンガーは、波動関数 Ψ の物理的な意味を問われた際に、「知らない」と答えたそうである。古典力学では粒子の質量は一点に集中している。しかし、波動関数値は空間全体に分布している。

シュレディンガー自身は、粒子が時間とともに粉々に分散してゆくというイメージで捉えていたようである。しかし、どう考えても粒子が粉々に分散してゆくというイメージはおかしい。

そこで出てきたのがボルンの確率解釈である。ボルンは、粒子の位置は決定論的なものではなくて、確率論的なものと考えた。波動関数 $\Psi(x, t)$ の絶対値の自乗 $|\Psi(x, t)|^2$ が、空間の各位置における存在確率密度分布を与えると考えた。もちろん、 $|\Psi(x, t)|^2$ は確率密度分布であるから、全空間における積分が 1 になるように正規化されているものとする。

これは驚天動地の発想で、古典力学に馴染んできた当時の殆どの物理学者は理解できなかった。一般の物理学者だけでなく、大家のアインシュタインやシュレディンガーですら理解できなかった。しかし、これは理解できるかどうかの問題ではなくて、受け入れるかどうかの問題である。

物理学の役割は自然をあるがままに記述することであるから、確率解釈が例外なく実験結果を説明できる限り、受け入れるのが正しいと言えよう。現在では、確率解釈が正統的な解釈とされ、現在の量子力学はこの解釈に立脚している。確率解釈は、1 光子の 2 重スリット実験やマッハ・ツェンダー干渉計の実験結果を見るとよく理解できる。

2.5 演算子のエルミート性

本節以降でたびたび出て来る演算子のエルミート性について触れる。これは確率解釈を成り立たせるための重要な性質である。

シュレディンガーの方程式：

$$i\hbar \frac{\partial \Psi}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x, t) \right) \Psi = \hat{H}\Psi \quad (2.21)$$

の解 $\Psi(x, t)$ には、次の性質：

$$\int_{-\infty}^{\infty} |\Psi(x, t)|^2 dx = 1, \quad \lim_{x \rightarrow \pm\infty} \Psi(x, t) = 0, \quad \lim_{x \rightarrow \pm\infty} \left| \frac{\partial \Psi}{\partial x} \right| < \infty \quad (2.37)$$

を課す。

もしも、ある時刻 $t = t_0$ で

$$\int_{-\infty}^{\infty} |\Psi(x, t_0)|^2 dx = 1 \quad (2.38)$$

ならば、 $t > t_0$ においても 1 であり続ける。そのために、まず

$$N(t) = \int_{-\infty}^{\infty} |\Psi(x, t)|^2 dx, \quad \text{If } N(t_0) = 1, \text{ then } \frac{dN}{dt} = 0 \text{ for } t > t_0 \quad (2.39)$$

が成り立てばよい。このことを証明する。 $dN/dt = 0$ が言えるためには

$$\frac{dN}{dt} = \frac{d}{dt} \int_{-\infty}^{\infty} |\Psi(x,t)|^2 dx = \int_{-\infty}^{\infty} dx \left(\frac{\partial \Psi^*}{\partial t} \Psi + \Psi^* \frac{\partial \Psi}{\partial t} \right) = \int_{-\infty}^{\infty} dx \frac{i}{\hbar} \left((\hat{H}\Psi)^* \Psi - \Psi^* (\hat{H}\Psi) \right) \quad (2.40)$$

の右辺がゼロでなければならない。式(2.21)を用いて、式(2.40)の最右辺を書き直すと

$$\begin{aligned} \frac{dN}{dt} &= \int_{-\infty}^{\infty} dx \frac{i}{\hbar} \left(\left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x,t) \right) \Psi \right)^* \Psi - \Psi^* \left(\left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x,t) \right) \Psi \right) \\ &= -\frac{i\hbar}{2m} \int_{-\infty}^{\infty} dx \left(\frac{\partial^2 \Psi^*}{\partial x^2} \Psi - \Psi^* \frac{\partial^2 \Psi}{\partial x^2} \right) = -\frac{i\hbar}{2m} \int_{-\infty}^{\infty} dx \frac{\partial}{\partial x} \left(\frac{\partial \Psi^*}{\partial x} \Psi - \Psi^* \frac{\partial \Psi}{\partial x} \right) \\ &= -\frac{i\hbar}{2m} \left[\frac{\partial \Psi^*}{\partial x} \Psi - \Psi^* \frac{\partial \Psi}{\partial x} \right]_{-\infty}^{\infty} = 0 \end{aligned} \quad (2.41)$$

となる。ここで、式(2.37)の条件が用いられた。従って、ある時刻 $t=t_0$ で全空間での確率の総和が1であれば、 $t > t_0$ でも1であり続ける。

演算子 \hat{H} は、式(2.40)と式(2.41)より

$$\int_{-\infty}^{\infty} dx \left((\hat{H}\Psi)^* \Psi - \Psi^* (\hat{H}\Psi) \right) = 0 \quad (2.42)$$

を満足している。このような演算子をエルミート演算子という。エルミート性は波動関数の確率解釈が成り立つために、必須の性質である。

流体力学や電磁気学のような古典力学と量子力学のアナロジーを示す。表2.1に、以下の数式の各学問領域における役割を示す。

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \vec{J} = 0, \quad Q = \int_V \rho dV, \quad \frac{dQ}{dt} = \int_V \frac{\partial \rho}{\partial t} dV = - \int_V \nabla \cdot \vec{J} dV = - \int_S \vec{J} \cdot d\vec{S} \quad (2.43)$$

表 2.1 古典力学と量子力学のアナロジー

	流体力学	電磁気学	量子力学
ρ	質量密度	電荷密度	確率密度
\vec{J}	流量密度	電流密度	確率流密度
Q	質量	電荷	粒子の存在確率

2.6 波束の不確定性

量子力学には、ハイゼンベルグの不確定性原理と呼ばれるものがあり、位置と運動量の不確定性をとるとき

$$\Delta x \cdot \Delta p \geq \frac{\hbar}{2} \quad (2.44)$$

が成り立つ。このことを、フーリエ変換を通して導かれる波束の性質として説明する。 $\Psi(x,0)$ と $\Phi(k)$ の関係を

$$\Psi(x,0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \Phi(k) e^{ikx} dk, \quad \Phi(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \Psi(x,0) e^{-ikx} dx \quad (2.45)$$

とする。図2.1に、における波動関数とそのフーリエ変換を示す。図中の Δx と Δk は拡がり（不確定性）である。

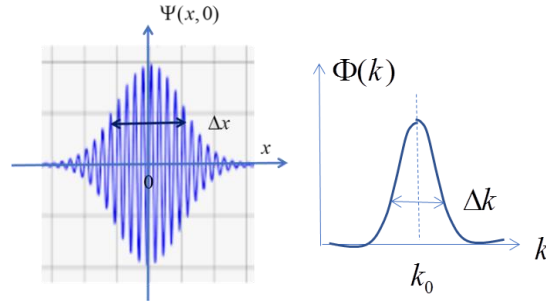


図 2.2 波動関数とそのフーリエ変換

簡単のために、 $\Phi(k)$ と k の関係を

$$\Phi(k) = \frac{1}{\Delta k}, \quad -\frac{\Delta k}{2} \leq k \leq \frac{\Delta k}{2} \quad (2.46)$$

とする。このとき

$$\Psi(x, 0) = \frac{1}{\sqrt{2\pi}} \int_{-\Delta k/2}^{\Delta k/2} \frac{1}{\sqrt{\Delta k}} e^{ikx} dk = \frac{1}{\sqrt{2\pi\Delta k}} \left[\frac{e^{ikx}}{ix} \right]_{-\Delta k/2}^{\Delta k/2} = \sqrt{\frac{\Delta k}{2\pi}} \frac{\sin\left(\frac{\Delta kx}{2}\right)}{\left(\frac{\Delta kx}{2}\right)} \quad (2.47)$$

となる。式 (2.46) と式 (2.47) を図にしたものを図 2.3 に示す。

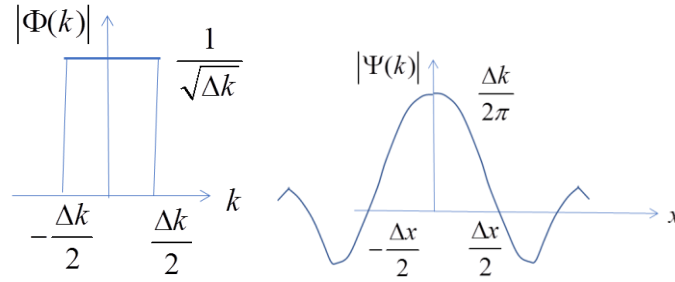


図 2.3 式 (2.46) と式 (2.47)

式 (2.47) より

$$\frac{\Delta x}{2} = \frac{2\pi}{\Delta k} \quad (2.48)$$

となる。これを書き直せば、式 (2.46) の仮定の下での不確定性：

$$\frac{\Delta x}{2} \frac{\Delta k}{2} = \pi \quad (2.49)$$

が求まる。

2.7 波束の時間進化

分散方程式が

$$\hbar\omega = \frac{k^2\hbar^2}{2m} + V \quad (2.20)$$

で与えられるとき、波動 $e^{i(kx - \omega(k)t)}$ はシュレディンガー方程式：

$$i\hbar \frac{\partial \Psi}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x, t) \right) \Psi = \hat{H} \Psi \quad (2.21)$$

の解であるので、 $t=0$ において式 (2.45) で与えられる波動関数 $\Psi(x, 0)$ は $t>0$ では

$$\Psi(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \Phi(k) e^{i(kx - \omega(k)t)} \quad (2.50)$$

で与えられることが容易に理解される。

2.8 フーリエ変換の復習と解釈

量子力学は波動関数が主役であるので、フーリエ変換が重要な役割を演ずる。そこで、フーリエ変換の基本的な事項をまとめて置く。

まず、関数 $\Psi(x)$, $-\infty < x < \infty$ と関数 $\Phi(k)$, $-\infty < k < \infty$ のフーリエ変換は次式で与えられる：

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \Phi(k) e^{ikx}, \quad \Phi(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx \Psi(x) e^{-ikx} \quad (2.51)$$

$\Psi(x)$ がディラックのデルタ関数：

$$\delta(x) = 0 \text{ when } x \neq 0 \text{ and } \int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (2.52)$$

とすると、式(2.51)より

$$\Phi(k) = \frac{1}{\sqrt{2\pi}} \quad (2.53)$$

であるので、次式を得る：

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dk e^{ikx} \quad (2.54)$$

パーセバルの定理は次式のように導かれる：

$$\begin{aligned} \int_{-\infty}^{\infty} dx \Psi^*(x) \Psi(x) &= \int_{-\infty}^{\infty} dx \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \Phi^*(k) e^{-ikx} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk' \Phi(k') e^{ik'x} \\ &= \int_{-\infty}^{\infty} dk \Phi^*(k) \int_{-\infty}^{\infty} dk' \Phi(k') \frac{1}{2\pi} \int_{-\infty}^{\infty} dx e^{i(k'-k)x} = \int_{-\infty}^{\infty} dk \Phi^*(k) \int dk' \Phi(k') \delta(k'-k) = \int_{-\infty}^{\infty} dk \Phi^*(k) \Phi(k) \end{aligned} \quad (2.55)$$

この式を整理すると

$$\int dx |\Psi(x)|^2 = \int dk |\Phi(k)|^2 \quad (2.56)$$

となる。

以上はフーリエ変換の復習であるが、以下に量子力学特有のフーリエ変換の解釈を述べる。すなわち、波数空間の運動量空間への変換である。式(2.2)：

$$p = \hbar k \quad (2.2)$$

により、波数 k を運動量 p に変換する。その結果

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int \frac{dp}{\hbar} \tilde{\Phi}(p) e^{ipx/\hbar}, \quad \tilde{\Phi}(p) = \frac{1}{\sqrt{2\pi}} \int dx \Psi(x) e^{-ipx/\hbar} \quad (2.57)$$

を得る。さらに

$$\tilde{\Phi}(p) \rightarrow \Phi_p(p) \sqrt{\hbar} \quad (2.58)$$

とすると

$$\Psi(x) = \frac{1}{\sqrt{2\pi\hbar}} \int dp \Phi_p(p) e^{ipx/\hbar}, \quad \Phi_p(p) = \frac{1}{\sqrt{2\pi\hbar}} \int dx \Psi(x) e^{-ipx/\hbar} \quad (2.59)$$

を得る。パーセバルの定理は

$$\int dx |\Psi(x)|^2 = \int dp |\Phi_p(p)|^2 \quad (2.60)$$

となる。

3次元の場合には

$$\Psi(\vec{x}) = \frac{1}{(2\pi\hbar)^{3/2}} \int d^3\vec{p} \Phi_p(\vec{p}) e^{i\vec{p}\cdot\vec{x}/\hbar}, \quad \Phi_p(\vec{p}) = \frac{1}{(2\pi\hbar)^{3/2}} \int d^3\vec{x} \Psi(\vec{x}) e^{-i\vec{p}\cdot\vec{x}/\hbar} \quad (2.61)$$

$$\delta^3(\vec{x} - \vec{x}') = \frac{1}{(2\pi)^3} \int d^3\vec{k} e^{i\vec{k}\cdot(\vec{x}-\vec{x}')} \quad (2.62)$$

$$\int d^3\vec{x} |\Psi(\vec{x})|^2 = \int d^3\vec{P} |\Phi_p(\vec{P})|^2 \quad (2.63)$$

が成り立つ。

2.9 演算子の期待値

粒子が $(x, x+dx)$ にある期待値 $\langle x \rangle$ は

$$\langle x \rangle = \int_{-\infty}^{\infty} x \Psi^*(x) \Psi(x) dx \quad (2.64)$$

で与えられる。同様に粒子の運動量が $(p, p+dp)$ にある期待値は、運動量空間で

$$\langle \hat{p} \rangle = \int_{-\infty}^{\infty} p \Psi^*(p) \Psi(p) dp \quad (2.65)$$

で与えられる。

以下に $\langle \hat{p} \rangle$ の座標空間での表現を求める：

$$\begin{aligned} \langle \hat{p} \rangle &= \int p \Phi_p^*(p) \Phi_p(p) dp = \int p dp \int \frac{dx'}{\sqrt{2\pi\hbar}} e^{\frac{ipx'}{\hbar}} \Psi^*(x') \int \frac{dx}{\sqrt{2\pi\hbar}} e^{-\frac{ipx}{\hbar}} \Psi(x) \\ &= \int dp \int \frac{dx'}{\sqrt{2\pi\hbar}} e^{\frac{ipx'}{\hbar}} \Psi^*(x') \int \frac{dx}{\sqrt{2\pi\hbar}} p e^{-\frac{ipx}{\hbar}} \Psi(x) = \int dp \int \frac{dx'}{\sqrt{2\pi\hbar}} e^{\frac{ipx'}{\hbar}} \Psi^*(x') \int \frac{dx}{\sqrt{2\pi\hbar}} \left(-\frac{\hbar}{i} \frac{\partial}{\partial x} e^{-\frac{ipx}{\hbar}} \right) \Psi(x) \\ &= \int dp \int \frac{dx'}{\sqrt{2\pi\hbar}} e^{\frac{ipx'}{\hbar}} \Psi^*(x') \int \frac{dx}{\sqrt{2\pi\hbar}} e^{-\frac{ipx}{\hbar}} \left(\frac{\hbar}{i} \frac{\partial}{\partial x} \Psi(x) \right) = \int dx' \Psi^*(x') \int dx \left(\frac{\hbar}{i} \frac{\partial}{\partial x} \Psi(x) \right) \frac{1}{2\pi\hbar} \int dp e^{\frac{ip(x'-x)}{\hbar}} \\ &= \int dx' \Psi^*(x') \int dx \left(\frac{\hbar}{i} \frac{\partial}{\partial x} \Psi(x) \right) \delta(x' - x) = \int dx \Psi^*(x) \left(\frac{\hbar}{i} \frac{\partial}{\partial x} \right) \Psi(x) = \int dx \Psi^*(x) \hat{p} \Psi(x) \end{aligned} \quad (2.66)$$

式(2.66)を整理すると

$$\langle \hat{p} \rangle = \int dx \Psi^*(x) \hat{p} \Psi(x) \quad (2.67)$$

が求まる。

一般に任意の演算子 \hat{Q} に対して、期待値 $\langle \hat{Q} \rangle$ は次式で与えられる：

$$\langle \hat{Q} \rangle = \int dx \Psi^*(x) \hat{Q} \Psi(x) \quad (2.68)$$

例；運動エネルギー演算子 \hat{T} ：

$$\hat{T} = \frac{\hat{p}^2}{2m} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \quad (2.69)$$

について考えて見よう。

座標空間で考えると、 \hat{T} の期待値 $\langle \hat{T} \rangle$ は

$$\langle \hat{T} \rangle = \int_{-\infty}^{\infty} dx \Psi^*(x, t) \hat{T} \Psi(x, t) = \int_{-\infty}^{\infty} dx \Psi^*(x, t) \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \right) \Psi(x, t) = \int_{-\infty}^{\infty} dx \frac{\hbar^2}{2m} \frac{\partial \Psi^*(x, t)}{\partial x} \frac{\partial \Psi(x, t)}{\partial x} \quad (2.70)$$

となる。運動量空間で考えると

$$\langle \hat{T} \rangle = \int_{-\infty}^{\infty} dp \frac{p^2}{2m} |\Psi_p(p, t)|^2 = \int_{-\infty}^{\infty} dp \frac{p^2}{2m} \Psi_p^*(p, t) \Psi_p(p, t) \quad (2.71)$$

となる。

2.10 演算子の時間進化

演算子 \hat{Q} の時間進化（発展）を考えて見よう。まず

$$\frac{d\langle\hat{Q}\rangle}{dt} = \frac{d}{dt} \int_{-\infty}^{\infty} dx \psi^* \hat{Q} \psi = \int_{-\infty}^{\infty} dx \left(\frac{\partial \psi^*}{\partial t} \hat{Q} \psi + \psi^* \hat{Q} \frac{\partial \psi}{\partial t} \right) = \int_{-\infty}^{\infty} dx \left(\frac{i}{\hbar} (\hat{H} \Psi)^* \hat{Q} \psi - \frac{i}{\hbar} \psi^* \hat{Q} \hat{H} \psi \right) \quad (2.72)$$

さらに、式(2.42)で与えられる \hat{H} のエルミート性により

$$\begin{aligned} i\hbar \frac{d\langle\hat{Q}\rangle}{dt} &= \int_{-\infty}^{\infty} dx \left(\psi^* \hat{Q} \hat{H} \psi - (\hat{H} \Psi)^* \hat{Q} \psi \right) = \int_{-\infty}^{\infty} dx \left(\psi^* \hat{Q} \hat{H} \psi - \psi^* \hat{H} \hat{Q} \psi \right) \\ &= \int_{-\infty}^{\infty} dx \psi^* (\hat{Q} \hat{H} - \hat{H} \hat{Q}) \psi = \int_{-\infty}^{\infty} dx \psi^* [\hat{Q}, \hat{H}] \psi \end{aligned} \quad (2.73)$$

を得る。ここで

$$[\hat{Q}, \hat{H}] = \hat{Q} \hat{H} - \hat{H} \hat{Q} \quad (2.74)$$

とする。

式(2.73)より

$$i\hbar \frac{d\langle\hat{Q}\rangle}{dt} = \langle [\hat{Q}, \hat{H}] \rangle \quad (2.75)$$

を得る。 \hat{Q} と \hat{H} が可換であれば、式(2.75)の右辺は0になる。物理量がハミルトン演算子と可換であれば、その物理量の平均値は時間変化しない。

2.11 オブザーバブル（観測可能な物理量）とハミルトニアン

すべての演算子には固有値があり、その固有値が物理量の期待値である。演算子 \hat{Q} がエルミート演算子とする。すなわち、式(2.42)により

$$\int dx \Psi_1^* \hat{Q} \Psi_2 = \int dx (\hat{Q} \Psi_1)^* \Psi_2 \quad (2.76)$$

とする。

ここで、関数 Ψ_1 と Ψ_2 の内積 (Ψ_1, Ψ_2) を次式で定義する：

$$(\Psi_1, \Psi_2) = \int dx \Psi_1^* \Psi_2 \quad (2.77)$$

内積には、つぎの性質がある。スカラー a に対して

$$(a\Psi_1, \Psi_2) = a^* (\Psi_1, \Psi_2), \quad (\Psi_1, a\Psi_2) = a (\Psi_1, \Psi_2) \quad (2.78)$$

となる。

エルミート演算子 \hat{Q} の Ψ での期待値を考えると

$$\langle\hat{Q}\rangle_{\Psi} = \int dx \Psi^* \hat{Q} \Psi = \int dx (\hat{Q} \Psi)^* \Psi = (\Psi, \hat{Q} \Psi) = (\hat{Q} \Psi, \Psi) \quad (2.79)$$

を得る。 \hat{Q} に関して二つの性質、すなわち、期待値 $\langle\hat{Q}\rangle_{\Psi}$ と固有値 q が実数であるという性質がある。まず、前者は以下のように証明される：

$$\left(\langle\hat{Q}\rangle_{\Psi} \right)^* = \left(\int dx \Psi^* \hat{Q} \Psi \right)^* = \int dx \Psi (\hat{Q} \Psi)^* = \int dx (\hat{Q} \Psi)^* \Psi = \int dx \Psi^* \hat{Q} \Psi = \langle\hat{Q}\rangle_{\Psi} \quad (2.80)$$

後者は

$$\langle\hat{Q}\rangle_{\Psi} = \int dx \Psi^* \hat{Q} \Psi = q \int dx \Psi^* \Psi = q \int dx |\Psi|^2 \quad (2.81)$$

により示される。

2.12 エルミート演算子の固有値：スペクトル理論

エルミート演算子 \hat{Q} の固有関数と固有値を考える：

$$\hat{Q}\Psi_1 = q_1\Psi_1, \quad \hat{Q}\Psi_2 = q_2\Psi_2, \quad \dots \quad (2.82)$$

固有関数は正規直交性：

$$\int dx \Psi_i^* \Psi_j = \delta_{ij} \quad (2.83)$$

を満たすように構成できる。この証明は以下のように与えられる。 $q_i \neq q_j$ とすると

$$\int dx \Psi_i^* \hat{Q}\Psi_j = q_j \int dx \Psi_i^* \Psi_j, \quad \int dx (\hat{Q}\Psi_i)^* \Psi_j = q_i \int dx \Psi_i^* \Psi_j \rightarrow (q_j - q_i) \int dx \Psi_i^* \Psi_j = 0 \quad (2.84)$$

直交性を利用すると、演算子 \hat{Q} の固有関数は基底関数となって、任意の Ψ を固有関数の重ね合わせとして記述できる：

$$\begin{aligned} \Psi(x, t) &= \alpha_1(t)\Psi_1(x) + \alpha_2(t)\Psi_2(x) + \dots = \sum_i \alpha_i(t)\Psi_i(x), \\ \alpha_i(t) &= (\Psi_i, \Psi) = \int dx \Psi_i^*(x)\Psi(x, t) \end{aligned} \quad (2.85)$$

展開係数 $\alpha_i(t)$ には、つぎの性質がある：

$$\int dx |\Psi|^2 = \int dx \left(\sum_i \alpha_i \Psi_i \right)^* \left(\sum_j \alpha_j \Psi_j \right) = \sum_i \sum_j \alpha_i^* \alpha_j \int dx \Psi_i^* \Psi_j = \sum_i \sum_j \alpha_i^* \alpha_j \delta_{ij} = \sum_i |\alpha_i|^2 = 1 \quad (2.86)$$

ここで、量子力学において重要な概念である「測定」について説明したい。一般的状態 Ψ において演算子 \hat{Q} を測定すると、ある状態 Ψ_i の固有値 q_i になる。 q_i を測定する確率は $P_i = |\alpha_i|^2 = |(\Psi_i, \Psi)|^2$ である。測定すると系の状態は $\Psi = \Psi_i$ となる（波動関数の消滅）。このことは、演算子 \hat{Q} の期待値 $\langle \hat{Q} \rangle_\Psi$ に関する次式により理解されよう：

$$\begin{aligned} \Psi = \sum_i \alpha_i \Psi_i \rightarrow \langle \hat{Q} \rangle_\Psi &= \int dx \left(\sum_i \alpha_i \Psi_i \right)^* \hat{Q} \left(\sum_j \alpha_j \Psi_j \right) = \sum_i \sum_j \alpha_i^* \alpha_j \int dx \Psi_i^* \hat{Q} \Psi_j \\ &= \sum_i \sum_j \alpha_i^* \alpha_j q_j \int dx \Psi_i^* \Psi_j = \sum_i \sum_j \alpha_i^* \alpha_j q_j \delta_{ij} = \sum_i |\alpha_i|^2 q_i \end{aligned} \quad (2.87)$$

2.13 量子力学から古典力学へ（エーレンフェストの定理）

以上は古典力学からの量子力学の導入であったが、本節では量子力学から見る古典力学の位置づけを述べたい。

いわゆるエーレンフェストの定理と呼ばれるもので、ポテンシャル V の下での粒子の運動の状態を表す波動関数を Ψ とするとき、ニュートンの運動方程式によく似た次式が成り立つ：

$$m \frac{d^2 \langle x \rangle}{dt^2} = - \left\langle \frac{\partial V}{\partial x} \right\rangle, \quad \text{where } \langle x \rangle = \int \Psi^* x \Psi dx, \quad \left\langle \frac{\partial V}{\partial x} \right\rangle = \int \Psi^* \frac{\partial V}{\partial x} \Psi dx \quad (2.88)$$

以下のように証明される。まず、 Ψ が無限遠で 0 になることを用いると

$$\begin{aligned} \frac{d^2 \langle x \rangle}{dt^2} &= \frac{d^2}{dt^2} \int \Psi^* x \Psi dx = \frac{d}{dt} \int \left[\frac{\partial \Psi^*}{\partial t} x \Psi + \Psi^* x \frac{\partial \Psi}{\partial t} \right] dx \\ &= \frac{1}{i\hbar} \frac{d}{dt} \int \left[- \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V \right) \Psi^* x \Psi + \Psi^* x \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V \right) \Psi \right] dx \\ &= \frac{1}{i\hbar} \frac{d}{dt} \int \left[- \left(-\frac{\hbar^2}{2m} \frac{\partial^2 \Psi^*}{\partial x^2} \right) x \Psi + \Psi^* x \left(-\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} \right) \right] dx = -\frac{i\hbar}{2m} \frac{d}{dt} \int \left[\frac{\partial^2 \Psi^*}{\partial x^2} x \Psi - \Psi^* x \frac{\partial^2 \Psi}{\partial x^2} \right] dx \end{aligned} \quad (2.89)$$

となる。さらに

$$\int \frac{\partial^2 \Psi^*}{\partial x^2} x \Psi dx = - \int \frac{\partial \Psi^*}{\partial x} \frac{\partial x \Psi}{\partial x} dx = \int \Psi^* \frac{\partial^2 x \Psi}{\partial x^2} dx = \int \left[2 \Psi^* \frac{\partial \Psi}{\partial x} + \Psi^* x \frac{\partial^2 \Psi}{\partial x^2} \right] dx \quad (2.90)$$

である。これを用いると

$$m \frac{d^2 \langle x \rangle}{dt^2} = -i\hbar \frac{d}{dt} \int \psi^* \frac{\partial \psi}{\partial x} dx = -i\hbar \frac{d}{dt} \left[\int \frac{\partial \psi^*}{\partial t} \frac{\partial \psi}{\partial x} + \psi^* \frac{\partial}{\partial x} \frac{\partial \psi}{\partial t} \right] dx \quad (2.91)$$

となる。再度、シュレディンガー方程式を用いると

$$m \frac{d^2 \langle x \rangle}{dt^2} = - \int \psi^* \frac{\partial V}{\partial x} \psi dx \quad (2.92)$$

を得る。右辺は $-\langle \partial V / \partial x \rangle$ に他ならないので、式(2.88)が求まる。

2.14 モードに分解されたシュレディンガーの方程式

ここでは、2.12節のスペクトル理論に従う説明をする。次章の量子計算の章では、量子力学に馴染んでいない読者に向けた古典力学的な説明をする。

式(2.85)によると、任意の Ψ を式(2.82)で与えられるエルミート演算子であるハミルトニアン \hat{H} の固有値問題：

$$\hat{H}Y_0 = E_0 Y_0, \quad \hat{H}Y_1 = E_1 Y_1, \quad \dots, \quad \hat{H}Y_{N-1} = E_{N-1} Y_{N-1} \quad (2.93)$$

の固有関数 $\Psi_0, \Psi_1, \dots, \Psi_{N-1}$ の線形結合として記述できる（演算子 \hat{Q} を \hat{H} ，固有値 q_0, q_1, \dots, q_{N-1} をエネルギー固有値 E_0, E_1, \dots, E_{N-1} としている。また、 N は無限大でもよい。）：

$$\begin{aligned} \Psi(x, t) &= \alpha_0(t) \Psi_0(x) + \alpha_1(t) \Psi_1(x) + \dots + \alpha_{N-1}(t) \Psi_{N-1}(x) = \sum_{i=0}^{N-1} \alpha_i(t) \Psi_i(x), \\ \alpha_i(t) &= (\Psi_i, \Psi) = \int dx \Psi_i^*(x) \Psi(x, t) \end{aligned} \quad (2.85)$$

これをシュレディンガー方程式：

$$i\hbar \frac{\partial \Psi}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x, t) \right) \Psi = \hat{H} \Psi \quad (2.21)$$

に代入して、固有関数の正規直交性：

$$\int dx \Psi_i^* \Psi_j = \delta_{ij} \quad (2.83)$$

を用いると

$$i\hbar \frac{d\alpha_n(t)}{dt} = E_n \alpha_n(t), \quad n = 0, 1, \dots, N-1 \quad (2.94)$$

という固有モードに分解されたシュレディンガーの方程式を得る。

ハミルトニアン \hat{H} が時間の関数でない場合は、エネルギー固有値 E_0, E_1, \dots, E_{N-1} も時間の関数でない。従って、常微分方程式(2.94)は簡単に解ける。すなわち

$$\alpha_n(t) = \exp\left(-\frac{i}{\hbar} E_n t\right) \alpha_n(0), \quad n = 0, 1, \dots, N-1 \quad (2.95)$$

となる。 $\exp(-(i/\hbar)E_n t)$ は時間進化演算子で

$$\left| \exp(-(i/\hbar)E_n t) \right|^2 = 1 \quad (2.96)$$

であるので

$$|\alpha_n(t)|^2 = |\alpha_n(0)|^2 \quad (2.97)$$

となる。すなわち、各状態 Ψ_n を与える各モード n の確率は、それぞれに一定で時間変化しないことになる。測定時に得られるモード n の確率は $|\alpha_n(0)|^2$ である。すなわち、同じ測定を多数回繰り返すと、モード n の状態 Ψ_n が測定される確率は $|\alpha_n(0)|^2$ である。

時間間隔 Δt が十分に小さければ、式(2.95)式は次式で近似される：

$$\begin{aligned}
\alpha_n(t_m) &= \alpha_n(m\Delta t) = \exp\left(-\frac{i}{\hbar} E_n \Delta t\right) \alpha_n((m-1)\Delta t) \\
&= \exp\left(-\frac{i}{\hbar} E_n m\Delta t\right) \alpha_n(0), \quad n = 0, 1, \dots, N-1
\end{aligned} \tag{2.98}$$

ハミルトニアン \hat{H} が時間の関数でも、時間間隔 Δt が十分に小さければ、以下の式で近似されよう：

$$\begin{aligned}
\alpha_n(t_m) &= \alpha_n(m\Delta t) = \exp\left(-\frac{i}{\hbar} E_n((m-1)\Delta t)\right) \alpha_n((m-1)\Delta t) \\
&= \exp\left(-\frac{i}{\hbar} E_n((m-1)\Delta t)\right) \exp\left(-\frac{i}{\hbar} E_n((m-2)\Delta t)\right) \cdots \exp\left(-\frac{i}{\hbar} E_n(0)\right) \alpha_n(0), \quad n = 0, 1, \dots, N-1
\end{aligned} \tag{2.99}$$

式 (2.94) を行列形式に書くと

$$i\hbar \begin{pmatrix} d\alpha_0(t)/dt \\ d\alpha_1(t)/dt \\ \vdots \\ d\alpha_{N-1}(t)/dt \end{pmatrix} = \begin{pmatrix} E_0 & 0 & \cdots & 0 \\ 0 & E_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & E_{N-1} \end{pmatrix} \begin{pmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{N-1}(t) \end{pmatrix}, \quad n = 0, 1, \dots, N-1 \tag{2.100}$$

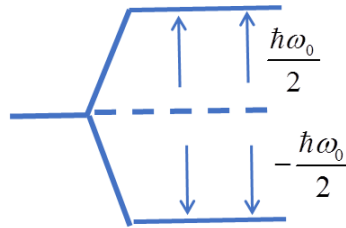
となる。

以上の議論は、1 電子の運動などのような 1 量子に N 個の運動モード（自由電子の場合は $N = \text{infinite}$ ，原子の中の電子の場合は $N = \text{finite}$ ）がある場合に、それを個々のモードに分解する場合の議論である。量子計算機の場合には、0 と 1 という 2 個のモード（状態）を持った多数の量子の全体の状態が議論される。個々の量子の $(0,1)$ の重ね合わせ状態は、全体の N 量子全体では 2^N 個の $(0,0,0,\dots,0,0), (1,0,0,\dots,0,0), (0,1,0,\dots,0,0), \dots, (0,1,1,\dots,1,1), (1,1,1,\dots,1,1)$ の重ね合わせ状態になる。これらの状態に最初から番号付けをすると、括弧内の 2 進数を 10 進数に読み替えた $0, 1, 2, \dots, 2^N - 2, 2^N - 1$ となる。

2.15 2 準位原子の問題（中山 茂，量子アルゴリズム，技報堂出版，第 1 版，（2014），p. 94）

(a) シュレディンガー方程式

図 2.4 に示されるように、エネルギー準位が 2 準位ある原子系を考える。本節以降、ディラックが導入した記法を用いることにする。全体の状態を $|\psi\rangle$ とし、二つの状態を $|0\rangle$ と $|1\rangle$ とする。



エネルギー準位

図 2.4 2 準位の原子系

式 (2.21) のシュレディンガー方程式は

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = \hat{H}|\psi(t)\rangle \tag{2.101}$$

と書かれる。状態 $|\psi(t)\rangle$ を

$$|\psi(t)\rangle = a(t)|0\rangle + b(t)|1\rangle = \begin{pmatrix} a(t) \\ b(t) \end{pmatrix} \quad (2.102)$$

と書くことにする。これは、2 モード（自由度）の場合の式 (2.85) に相当する。状態 $|\psi(t)\rangle$ は 2 次元の列ベクトル $(a(t) \ b(t))^T$ と同等と考えてよい。

式 (2.100) を参考にして、2 準位のハミルトニアン演算子を

$$\hat{H} = \begin{pmatrix} -\frac{\hbar\omega_0}{2} & 0 \\ 0 & \frac{\hbar\omega_0}{2} \end{pmatrix} \quad (2.103)$$

と、行列形式で書いたシュレディンガー方程式は

$$i\hbar \begin{pmatrix} \frac{da(t)}{dt} \\ \frac{db(t)}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{\hbar\omega_0}{2} & 0 \\ 0 & \frac{\hbar\omega_0}{2} \end{pmatrix} \begin{pmatrix} a(t) \\ b(t) \end{pmatrix} \quad (2.104)$$

となる。この微分方程式の解は

$$a(t) = a(0)e^{i\frac{\omega_0}{2}t}, \quad b(t) = b(0)e^{-i\frac{\omega_0}{2}t} \quad (2.105)$$

で与えられる。 $|a(t)|^2$ と $|b(t)|^2$ は測定時に $|0\rangle$ と $|1\rangle$ が観測される確率であるから

$$|a(t)|^2 + |b(t)|^2 = |a(0)|^2 + |b(0)|^2 = 1 \quad (2.106)$$

を満たさねばならない。

(b) レーザーによる操作

2 準位原子系の状態はレーザーによる操作が可能である。角周波数 ω のレーザー光と原子の相互作用の強さを Ω とする。このときのハミルトニアンは

$$\hat{H} = \begin{pmatrix} -\frac{\hbar\omega_0}{2} & 0 \\ 0 & \frac{\hbar\omega_0}{2} \end{pmatrix} + \begin{pmatrix} 0 & -\frac{\hbar\Omega}{2}e^{i\omega t} \\ -\frac{\hbar\Omega}{2}e^{-i\omega t} & 0 \end{pmatrix} = \begin{pmatrix} -\frac{\hbar\omega_0}{2} & -\frac{\hbar\Omega}{2}e^{i\omega t} \\ -\frac{\hbar\Omega}{2}e^{-i\omega t} & \frac{\hbar\omega_0}{2} \end{pmatrix} \quad (2.107)$$

で与えられる。従って、この場合のシュレディンガー方程式は

$$i\hbar \begin{pmatrix} \frac{da(t)}{dt} \\ \frac{db(t)}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{\hbar\omega_0}{2} & -\frac{\hbar\Omega}{2}e^{i\omega t} \\ -\frac{\hbar\Omega}{2}e^{-i\omega t} & \frac{\hbar\omega_0}{2} \end{pmatrix} \begin{pmatrix} a(t) \\ b(t) \end{pmatrix} \quad (2.108)$$

となる。

変数変換：

$$a(t) = a'(t)e^{i\frac{\omega}{2}t}, \quad b(t) = b'(t)e^{-i\frac{\omega}{2}t} \quad (2.109)$$

を行うと、微分方程式は

$$\frac{da'(t)}{dt} = \frac{i}{2}[(\omega_0 - \omega) + \Omega b'(t)], \quad \frac{db'(t)}{dt} = \frac{i}{2}[-(\omega_0 - \omega) + \Omega a'(t)] \quad (2.110)$$

となる。2 準位間のエネルギーと光のエネルギーが同じ、すなわち $\hbar\omega_0 = \hbar\omega$ とすると

$$\frac{da'(t)}{dt} = i\frac{\Omega}{2}b'(t), \quad \frac{db'(t)}{dt} = i\frac{\Omega}{2}a'(t) \quad (2.111)$$

を得る。変形すると

$$\frac{d^2 a'(t)}{dt^2} = -\frac{\Omega^2}{4} a'(t), \quad \frac{d^2 b'(t)}{dt^2} = -\frac{\Omega^2}{4} b'(t) \quad (2.112)$$

となる。これを解いて

$$a'(t) = a'(0) \cos \frac{\Omega}{2} t + ib'(0) \sin \frac{\Omega}{2} t, \quad b'(t) = b'(0) \cos \frac{\Omega}{2} t + ia'(0) \sin \frac{\Omega}{2} t \quad (2.113)$$

を得る。

変数を元に戻すと

$$|\psi(t)\rangle = \left[a(0) \cos \frac{\Omega}{2} t + ib(0) \sin \frac{\Omega}{2} t \right] e^{-i\frac{\omega_0}{2}t} |0\rangle + \left[b(0) \cos \frac{\Omega}{2} t + ia(0) \sin \frac{\Omega}{2} t \right] e^{i\frac{\omega_0}{2}t} |1\rangle \quad (2.114)$$

グローバルな指数関数の位相を無視すると

$$|\psi(t)\rangle = \left[a(0) \cos \frac{\Omega}{2} t + ib(0) \sin \frac{\Omega}{2} t \right] |0\rangle + \left[b(0) \cos \frac{\Omega}{2} t + ia(0) \sin \frac{\Omega}{2} t \right] |1\rangle \quad (2.115)$$

光をあてる時間や位相を変えると、 $|0\rangle$ と $|1\rangle$ の任意の重ね合わせが可能である。

(c) ラビ振動

原子の初期状態として、基底状態 $|0\rangle$ にあるとすると、 $a(0)=1$ 、 $b(0)=0$ である。光を原子に当て続けると、時刻 t の原子の状態と確率は

$$\begin{aligned} \text{状態: } |\psi(t)\rangle &= \cos\left(\frac{\Omega}{2}t\right)|0\rangle + i\sin\left(\frac{\Omega}{2}t\right)|1\rangle \\ \text{確率: } |a(t)|^2 &= \left|\cos\left(\frac{\Omega}{2}t\right)\right|^2, \quad |b(t)|^2 = \left|i\sin\left(\frac{\Omega}{2}t\right)\right|^2 \end{aligned} \quad (2.116)$$

すなわち、共鳴項を当て続けると、状態 $|0\rangle$ と $|1\rangle$ の間で確率が振動する（ラビ振動）。

2.16 ボーアとアインシュタインの論争

古典力学の世界観は決定論的である。古典力学に依れば、天体の運動はニュートンの運動理論により未来永劫まで予測可能ということになる。しかし、新しく登場した量子力学では、このような世界観は否定される。ボルンはシュレディンガー方程式の解である波動関数を、粒子の存在確率の波動であるとする確率解釈を提唱した。現在、この解釈は量子力学の根幹として広く受け入れられている。すなわち、量子力学の世界観は非決定論的である。

このことを巡り、当時の学会を二分する大論争が展開された。確率解釈に反対したのは、アインシュタインやシュレディンガーのような大家を筆頭とする当時の物理学会の権威者達であった。一方、支持者は若い人達だったと思うが、意外なことにように思えるが、支持派の筆頭は権威者のボーアであった。この論争はかなり長期に及んだが、次第に支持派が優勢になり現在では確率解釈は不動のものとなっている。

この大論争の渦中でアインシュタインは、有名な言葉を遺している。すなわち、「神はサイコロを振らない」という言葉である。この言葉は文学的な表現としては、簡潔簡明で魅力的なものであるが、科学的な表現としては問題があろう。しかし、アインシュタインの考え方を知る上で重要なものと考えられよう。

これに対してボーアは、「物理学の目的は、自然を正確に記述することにある」と述べたそうである。ボーアは、慎重に自分より年長者のアインシュタインのサイコロ説を批判したものと思われる。ボーア

の言は、「物理学の目的は、神の意志を伝えるものではない」と言ったのではなかろうか？著者にはそのように思えてならない。

アインシュタインのサイコロ説は、二重に正しくない。一つは、自然科学に神という宗教的権威を持ち込んでいることである。他は、量子力学が述べる本質的不規則性を、サイコロの持つ不規則性と混同していることである。正確に作られたサイコロを、正確に再現可能な方法で投げれば、結果は確率的ではなくて決定論的であろう。従って、サイコロの持つ不規則性は本質的なものではなくて、古典力学の近似に過ぎない。

量子力学のいう確率性は、自然の持つ本質的なものである。物質のスケールが極めて小さい超ミクロの原子や電子のような量子的世界では、確率性が重要な役割を担う。一方、我々が日常接するマクロの世界では、確率性の影響は小さく決定論的である。野球のボールやサッカーボールの位置は確率的ではない。厳密に言えば確率的であろうが、その確率密度分布はほとんど数学のデルタ関数であろう。

第2章の参考文献

[2-1] 小出昭一郎, 量子力学 (I), 裳華房 (1969).

[2-2] 小出昭一郎, 量子力学 (II), 裳華房 (1969).

第3章 量子計算の理論とは

3.1 量子計算の特徴

量子計算は、量子力学の理論に基づく計算方法である。量子計算の理論は矛盾のない数学理論であるが、特に独学の場合、簡単に理解できるとは言い難い。ベースとなっている量子力学そのものに、理解を難しくしている原因がある。

量子力学の特徴を挙げると

- (1) 量子性
- (2) 線型性
- (3) 2重性：量子は波でもあり粒子である
- (4) 非決定性：重ね合わせ状態
- (5) 不確定性
- (6) 遠距離相関：絡み合い

がある。中でも難解にしているのは、(4)の「重ね合わせ」と(6)「絡み合い」であろう。

直接的に隣り合うものの間、あるいは棒の両端のように間接的につながっているものの間に、絡み合い（相関）が発生すること事態は不思議でもなんでもない。不思議なのは重力や電磁気力のように、場を通して絡み合いが発生するのでもなく、ものどもの間に直接的に働く絡み合いである。何の縁もない地球上の粒子と月面上の粒子に絡み合いが発生することである。

計算それ自体は問題を表現するものの状態（情報）と答えを表すものの状態（情報）との間に、絡み合いを作り出すことであるから、計算理論は必然的に絡み合いを包含せざるを得ない。さらに、絡み合う二つのものの間の距離を変えても絡み合いは変わらないとする、伸びるとか縮むという力学的な絡み合いでは考えられないが、情報の世界ではあると言う事であろう。東京で時刻表示に1時間の差がある2個の時計の1個をロンドンに運んでも、2個の時計の差は1時間であろう。

(4)の「重ね合わせ」の理解し難さは、もっと本質的と思われる。そのために重ね合わせの説明から始めよう。一つの量子を例に取る。この量子を観測すると、あるときは1、ある時は0が観測されるとする。しかし、その観測値は完全に不規則ではなくて、多数の観測をしてそれぞれの頻度を取ると、1と0のそれぞれが観測される頻度は、観測回数を増やすとともに一定の値に収束するものとする。次に全く同じ粒子を多数個用意して、同時に観測し頻度を取ると1と0のそれぞれが観測される頻度は、量子の個数を増やすとともに一定の値に収束するものとする。すなわち、観測時に1が出る確率および2が出る確率を定義できることになる。

ここで、注意すべきことは、観測時に観察されるのは、必ず1か0のどちらかで0.3とか0.5のような1と0の中間の値ではないということである。それと、量子は観測されるとその値に留まるということである。量子は量子性により最小単位であるから、中間的な値は取れないから、当然のことであろう。

そこで、量子力学のディラックの記法に倣って、この量子の状態を $|\psi\rangle$ とし、観測時の0と1の状態を $|0\rangle$ と $|1\rangle$ とする。そして、 $|\psi\rangle$ は $|0\rangle$ と $|1\rangle$ の線形結合であるとする。すなわち

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \quad (3.1)$$

とする。 $|\psi\rangle$ はシュレディンガーの波動方程式の固有関数であるから、複素数値で二つの状態 $|0\rangle$ と $|1\rangle$ の確率を与えているはずである。そこで

$$|c_0|^2 + |c_1|^2 = 1 \quad (3.2)$$

と正規化できれば、 $|c_0|^2$ と $|c_1|^2$ が二つの状態 $|0\rangle$ と $|1\rangle$ の確率を与えていると考えられよう。

状態 $|\psi\rangle$ の情報は c_0 と c_1 に他ならないので、

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \quad (3.3)$$

と列ベクトルの形に書いて良いであろう。式 (3.1) を書き直すと

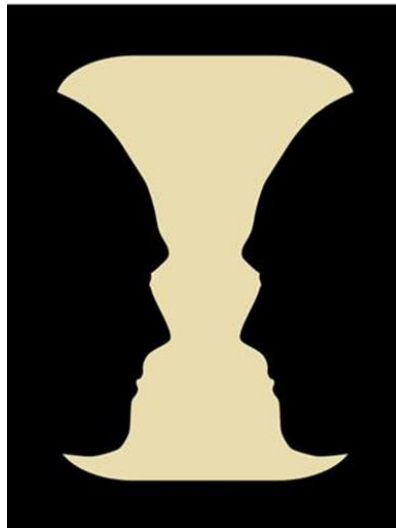
$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.4)$$

であるから

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.5)$$

である。式 (3.4) より、 $|\psi\rangle = (c_0 \ c_1)^T$ は、 $|0\rangle = (1 \ 0)^T$ と $|1\rangle = (0 \ 1)^T$ を基底とし、複素数値の座標を持つ 2 次元空間である。

式 (3.1) は量子計算機の本質に関わる「重ね合わせ」を数式化したものであるが、すんなり納得できない恐れがある。これに近いことが我々の身近にないように思えるが、実はそうではない。図 3.1 を見ていただきたい。



Rubinの壺の一例

図 3.1 Rubin の壺

<https://ja.wikipedia.org/wiki/%E3%83%AB%E3%83%93%E3%83%B3%E3%81%AE%E5%A3%BA>

この絵を見ると、壺と向き合う二人の顔という二つのイメージが描かれているが、一度に見えるのは壺と顔のどちらかである。従って、式 (3.1) で表現できると思うが、残念ながら基底の係数 c_0 と c_1 に妥当な意味づけができない。強いてこじつけると、視点が壺の上端にあるときは壺が見え、下端にあるときには顔が見えるので、視点が上にあるか下にあるかの確率に関連していると考えられないこともない。

また、図 3.2 に別の例を上げる。この図には柱と女性が描かれているが、図 3.1 の場合よりも複雑な見え方をする。柱だけ、女性だけ、柱と女性の両方が見えるなどである。



図 3.2 同様な他の例

<https://ja.wikipedia.org/wiki/%E3%83%AB%E3%83%93%E3%83%B3%E3%81%AE%E5%A3%BA>

人間の脳の働きは決定論的でないように思える。人間の感情は時々刻々変わる。あれを考えたり、これを考えたり、想念も絶えず変化する。思いがけない時に思いがけないひらめきがある。このようなことを考えると、脳の働きは気まぐれとも思え、非決定論的と考えたくなる。脳の持つ非決定性が創造力の源泉であるかも知れない。

そもそも、人間という存在自体が摩訶不思議のかたまりである。我々は他人が何を考えているか知ることはできない。しかし、推定することはできる。推定できなければ、共存できなくなる恐れがある。何かの機会に他人の真意を知る機会がある。他人の脳の状態と量子の状態には似ている所が多いように思える。

古典計算機に倣って、1 量子のことを 1 量子ビット, 1qubit (1 キュービット) と呼ぶ。1 量子ビットの特徴を古典計算機の 1 ビットおよび 1 確率ビットと比較してみよう。

- ・古典計算機の 1 ビットは確率 1 で 1 になり、確率 1 で 0 になる。
- ・確率ビットは 0 と 1 の中間の値を取ることができ、どの値になるかは確率で与えられる。
- ・量子ビットは観測すると 0 と 1 のどちらかになるが、観測しないときには、0 であり、1 であるという重なり合いの状態、どっちともいえない状態にある。

量子ビットの表現の仕方に、ブロッホ球と呼ばれるものがある。量子ビットの状態 $|\psi\rangle$ は数理的に式 (3.1) のように表現され、基底 $|0\rangle$ と $|1\rangle$ の係数 c_0 と c_1 は式 (3.2) を満たす。 $|\psi\rangle$ を書き直すと

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \quad (3.6)$$

と書けよう。ここで

$$0 \leq \theta \leq \pi, \quad 0 \leq \phi < 2\pi \quad (3.7)$$

とする。 θ と ϕ が半径 1 の図 3.3 の球面の緯角と経角とすると、 (θ, ϕ) と球面の一点が対応する。この球をブロッホ球と呼ぶ。式 (3.6) の右辺に $e^{i\phi}$ を掛けたものも式 (3.2) を満足するが、球面上に留まるばかりでなく確率が変わる訳でもないので無視する。古典ビットは点、確率ビットは $[0,1]$ の直線の自由度であるのに対して、量子ビットは半径 1 の球面の自由度を持っている。

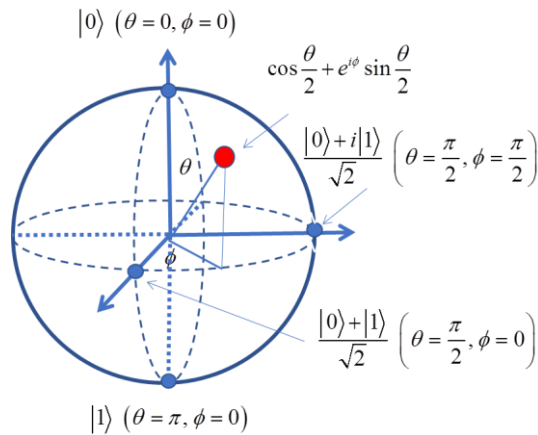


図 3.3 量子ビットのブロッホ球

量子計算の観点からは、量子ビットが実際に存在することが重要である。例えば、結晶格子上的磁気スピンには上向き下向きの2方向が重ね合わせを形成する。ジョセフソン素子（超伝導体の間に薄い絶縁体を挟んで弱く接合した「ジョセフソン接合」のトンネル効果を利用するリング状の素子）は、右向き左向きの電流の重ね合わせを形成する。

上で、量子力学の特徴として、(3) 2重性と(4) 非決定性を挙げたが、そのような物理現象を分かり易く見せてくれる実験結果について以下に述べる。

3.2 見ることができる量子的現象 1…ヤングの2重スリット

19世紀の初めにトーマス・ヤングが実験的に見出した光の波動性を示す現象である。1805年ころヤングは、コヒーレントな光源からの光を平行な2つのスリットを通す図3.4に示されるような実験を行った。そうすると、スリットを挟んで光源の反対側に設置されたスクリーンの上に干渉縞を生じることを発見した。これは光の波動性を示す現象である。量子力学では電子の流れも波動であるので、電子の流れでも干渉が見られる。同じような現象は水面波でも観察できる。

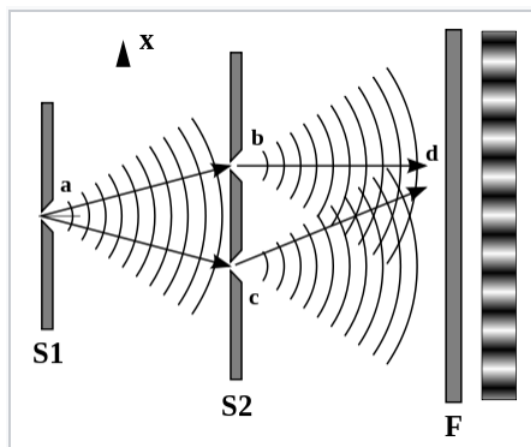


図 3.4 ヤングの2重スリット実験

片方のスリットをふさいでシングルスリットにすると、このような縞は現れないで、ダブルスリットの場合に縞が現れるのであるから、縞は二個のスリットを通過した2本の光線が干渉したためであることが分かる。要するに、シングルスリットの場合には、光はニュートンの言うように粒子として振る舞

い、ダブルスリットの場合には、光はフックやホイヘンスの言うように波動としての振る舞いが現れたものである。現在の我々にとっては、さして不思議と思われないが、よく考えればとても不思議である。

(1) 連続光による干渉

図 3.5 に示されるように座標を取る。

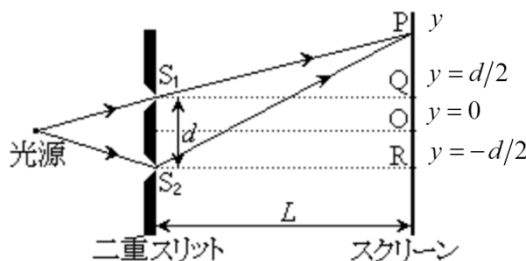


図 3.5 2 重スリットを通った光の干渉

スリット S_1 とスクリーン上の点 P を結ぶ光路の長さを PS_1 とすると

$$PS_1 = \left(L^2 + \left(y - \frac{d}{2} \right)^2 \right)^{\frac{1}{2}} = L \left(1 + \left(\frac{y - d/2}{L} \right)^2 \right) \quad (3.8)$$

となる。 L が大きいとして近似すると

$$PS_1 \approx L \left(1 + \frac{1}{2} \left(\frac{y - d/2}{L} \right)^2 \right) \quad (3.9)$$

となる。同様に考えると

$$PS_2 \approx L \left(1 + \frac{1}{2} \left(\frac{y + d/2}{L} \right)^2 \right) \quad (3.10)$$

を得る。

式 (3.9) と式 (3.10) より、光路差 Δ は

$$\Delta = PS_2 - PS_1 \approx L \frac{yd}{L^2} = \frac{yd}{L} \quad (3.11)$$

で与えられる。経路差が波長 λ の整数倍になると位相がそろうので強め合う。故に明線が現れる条件は

$$\frac{yd}{L} = n\lambda, \quad n = 1, 2, \dots \quad (3.12)$$

で与えられる。すなわち、明暗の縞ができることが理論的に示された。

(2) 1 光子による干渉

(1) では、スリットを通る光線を連続光と考えた。連続光は多数の光子の流れである。量子力学では光子を確率波の波束と考える。従って、干渉は 1 光子のレベルでも起きていると考えられよう。連続光の明るさを減らして行くと一つ一つの光子が連なった状態になる。このような場合にも干渉が起きるはずである。

浜松ホトニクスが行った実験結果を如何に示す。光子の代わりに電子を使った実験結果もある

(<https://www.youtube.com/watch?v=I9Ab8BLW3kA>, 二重スリットの実験(電子線バイプリズムによる干渉像)).

実験の概念図を図 3.6 に示す.

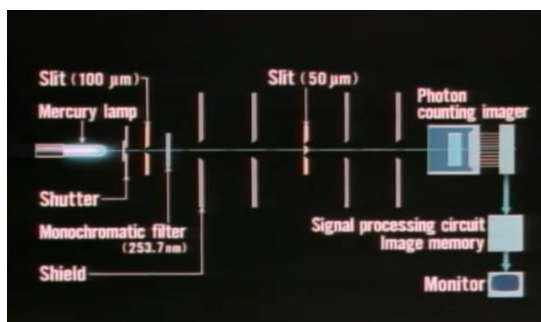


図 3.6 単一光子干渉実験概念図

図 3.7 に実験結果を示す. 見事にダブルスリットによる干渉を捉えている. 光の 2 重性の疑いの余地のない証拠である.

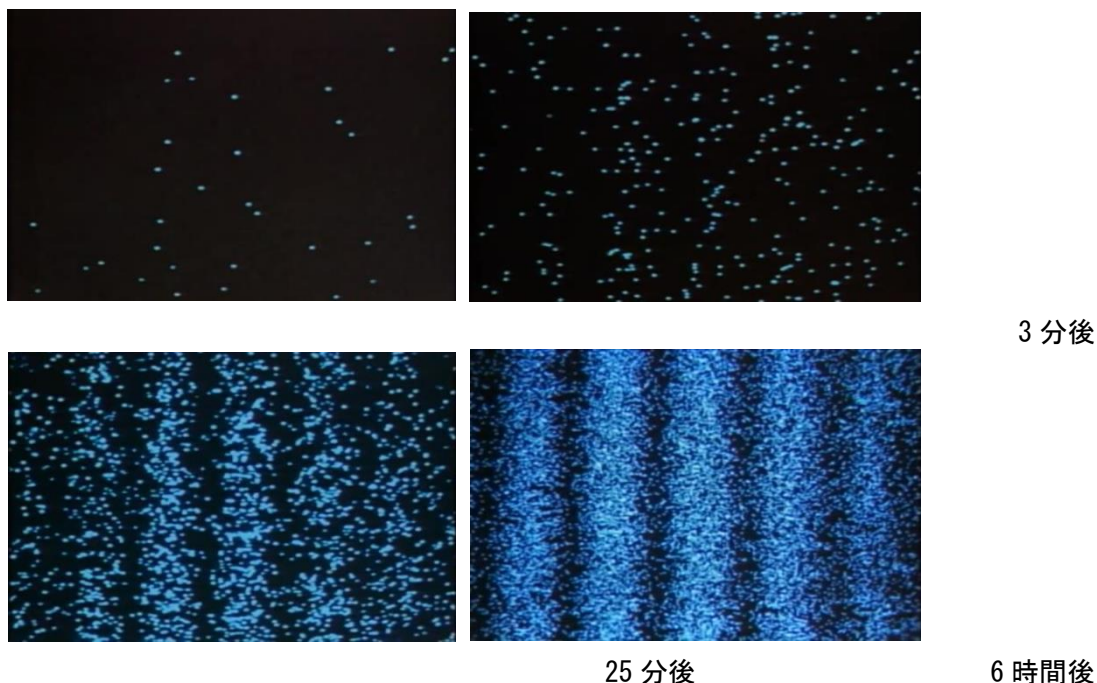


図 3.7 単一光子干渉実験

以上の実験結果を, シュレディンガー方程式との関連で説明する. 単一光子はこれ以上分割できない最小単位であるから, 単一光子はどちらかのスリットを通過する. 0.5 個の光子が物理的に同時に, 両方のスリットを通過する訳ではない. ボルンの確率解釈に示されるように, それぞれのスリットを通過する確率が 0.5 になるということである. このような非決定性 (確率性) が最も根源的な自然法則であることになる.

量子力学では, 光子は波束と考え, その波動関数 $\Psi(x, t)$ を次式とする :

$$\Psi(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \Phi(k) e^{i(kx - \omega(k)t)}, \quad \Phi(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx \Psi(x, 0) e^{-i(kx - \omega(k)t)} \quad (3.13)$$

従って

$$\Phi(k - k_0) = \delta(k - k_0) \rightarrow \Psi(x, t) = \frac{1}{\sqrt{2}} e^{i(k_0 x - \omega(k_0) t)} \quad (3.14)$$

と仮定すると取り扱いやすい。

確率波はスリットを波源として、直線的に進む1次元波で近似される。1スリットを波源とするレイセオリー近似で

$$\Psi(r, \theta, t) \approx \frac{f(\theta)}{r} e^{i(kr - \omega t)} \quad (3.15)$$

とする。小間隔の2重スリットを波源とするレイセオリー近似を

$$\Psi(r, \theta, t) \approx \frac{f(\theta_1)}{r_1} e^{i(kr_1 - \omega t)} + \frac{f(\theta_2)}{r_2} e^{i(kr_2 - \omega t)} \approx \frac{f(\theta_0)}{r_0} (e^{i(kr_1 - \omega t)} + e^{i(kr_2 - \omega t)}) \quad (3.16)$$

とし、図3.5に従って

$$r_1 = PS_1 \approx L \left(1 + \frac{1}{2} \left(\frac{y - d/2}{L} \right)^2 \right), \quad r_2 = PS_2 \approx L \left(1 + \frac{1}{2} \left(\frac{y + d/2}{L} \right)^2 \right), \quad \Delta = r_2 - r_1 = \frac{yd}{L} \quad (3.17)$$

とすると、スクリーン上では

$$\Psi(r, \theta, t) \approx \frac{f(\theta_0)}{r_0} (e^{i(kr_1 - \omega t)} + e^{i(kr_2 - \omega t)}) \approx \frac{f(\theta_0(y))}{r_0(y)} e^{i(kr_2 - \omega t)} \left(1 + e^{i k \frac{yd}{L}} \right) \quad (3.18)$$

となる。

故にスクリーン上の光子の粒子数密度分布は

$$\begin{aligned} |\Psi(r, \theta, t)|^2 &\approx \frac{f^2(\theta_0(y))}{r_0^2(y)} \left(1 + e^{-i k \frac{yd}{L}} \right) \left(1 + e^{i k \frac{yd}{L}} \right) = \frac{f^2(\theta_0(y))}{r_0^2(y)} \left(2 + e^{-i k \frac{yd}{L}} + e^{i k \frac{yd}{L}} \right) \\ &= 2 \frac{f^2(\theta_0(y))}{r_0^2(y)} \left(1 + \cos \left(k \frac{yd}{L} \right) \right) = 2 \frac{f^2(\theta_0(y))}{r_0^2(y)} \cos^2 \left(\frac{1}{2} k \frac{yd}{L} \right) \end{aligned} \quad (3.19)$$

となる。無限領域の場合には、全領域の粒子数が無限大になってしまい、正規化できないので、確率密度分布ではなくて粒子数密度分布とした。式(3.19)より、スクリーン上に粒子数分布の大小の縞ができることが理解されよう。

3.3 見ることができる量子的現象 2…マッハ・ツェンダー干渉計

マッハ・ツェンダー干渉計は、図3.8に示すようなもので、ビームスプリッターBS1で1本の光線を2本に分けて再びビームスプリッターBS2で合流させて、2本の光を干渉させてフェーズシフターで与えられる位相差を測定するものである。

BS2なし：光子～粒子（干渉なし） BS2あり：光子～波（干渉あり）

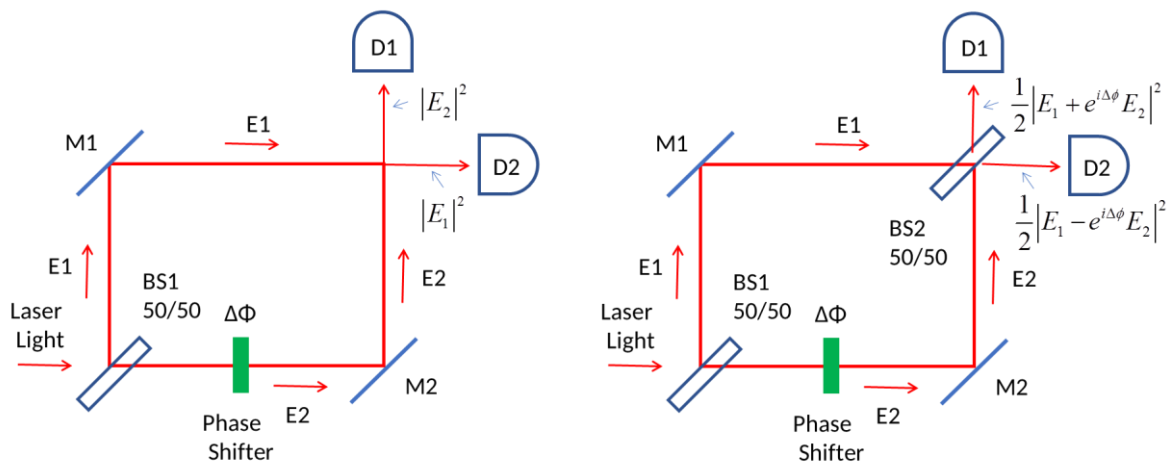


図 3.8 マッハ・ツェンダー干渉計の原理図

図 3.8 には、ディテクター $D1$ と $D2$ で検出される連続光の場合の干渉も示してある。

図 3.9 にマッハ・ツェンダー干渉計を使った計測例を示す。

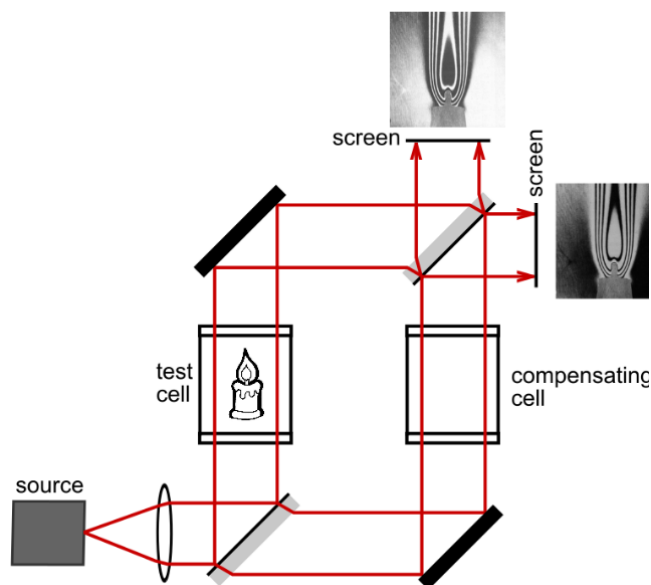


図 3.9 マッハ・ツェンダー干渉計の計測例

マッハ・ツェンダー干渉計を用いた、V. Jacques 等による 1 光子の興味ある実験結果を述べる。図 3.10 に実験の概念図を示す。この装置を使ってビームスプリッタ BS_{output} のある場合となしの場合の計測を行う。

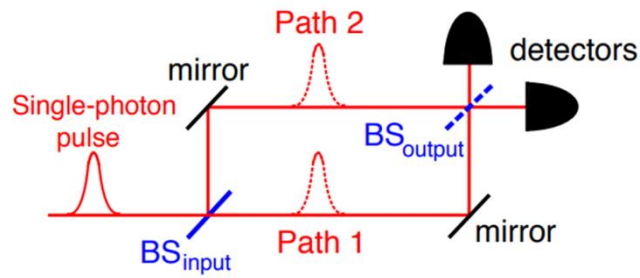


図 3.10 マッハ・ツェンダー干渉計を用いた 1 光子の実験

V. Jacques et al, Experimental realization of Wheeler's delayed-choice GedankenExperiment, Science 315, 966 (2007),

https://www.researchgate.net/publication/6501301_Experimental_Realization_of_Wheeler%27s_Delayed-Choice_Gedanken_Experiment

多数の単一光子を打ち込んだ実験結果が図 3.11 に示されている。BS_{output} なしの場合には、Path 1 を通った光子と Path 2 を通った光子が干渉を起こさずに、別々のディテクターに検出される。その時の光子の検出数が左側の図 (b) に示されている。50 : 50 の比率になっている。一方、BS_{output} ありの場合には、右側の図 (a) に示されているように、Path 1 を通った光子と Path 2 を通った光子数が正弦関数状に変化している。D1 の検出数が最大のときに D2 の検出数が最小になる。フェーズシフトが 1 波長分増えると逆になる。明らかに Path 1 を通った光子と Path 2 を通った光子が干渉している。BS_{input} に打ち込まれた単一光子が、BS_{output} を経てどのように D1 と D2 に届くかは後出の式 (3.28) に従う。

BS2なし：干渉なし

BS2あり：干渉あり

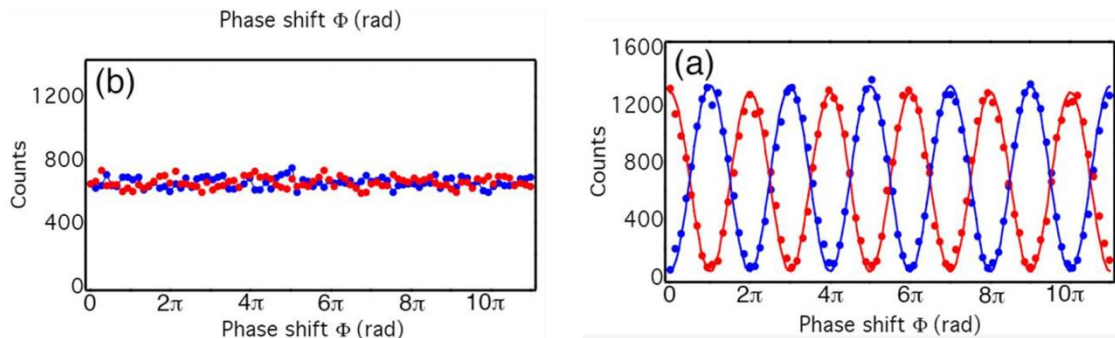


図 3.11 マッハ・ツェンダー干渉計による 1 光子干渉の実験結果

ヤングの 2 重スリットのところでも述べたように、この実験結果は極めて不可思議と言えよう。何故なら、単一光子はこれ以上分解できないものであるから、Path 1 と Path 2 をそれぞれ 0.5 光子が通過することはありえないからである。1 光子は必ず Path 1 と Path 2 のどちらかを通過するにも拘らず、連続光の場合と同じような干渉を起こすのである。

1 光子の実験で明かになったことは、1 光子でも干渉が起きることと、D1 と D2 の光子カウント数は光子の出現確率を与えていることである。従って、この干渉は物理的な波動の干渉ではなくて、確率波の干渉である。

マッハ・ツェンダー干渉計で起こる現象の数理について考察する。まず、連続光としての取り扱いであるが、BS ありとしてフェースシフトを $\Delta\phi$ とすると、D1 と D2 の光の強度 I_1 と I_2 は

$$2I_1 = |E_1 + e^{i\Delta\phi} E_1|^2 = E_1^* E_1 (1 + e^{-i\Delta\phi})(1 + e^{i\Delta\phi}) = E_1^* E_1 (2 + e^{i\Delta\phi} + e^{-i\Delta\phi}) = 2|E_1|^2 (1 + \cos(\Delta\phi)) \quad (3.20)$$

$$2I_2 = |E_1 - e^{i\Delta\phi} E_1|^2 = E_1^* E_1 (1 - e^{-i\Delta\phi})(1 - e^{i\Delta\phi}) = |E_1|^2 (2 - e^{i\Delta\phi} - e^{-i\Delta\phi}) = 2|E_1|^2 (1 - \cos(\Delta\phi)) \quad (3.21)$$

と求まる。

計算結果を図 3.12 に示す。

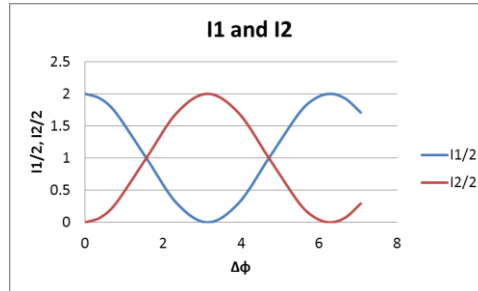


図 3.12 マッハ・ツェンダー干渉計による連続光の干渉

次に単一光子の場合について述べる。図 3.13 にマッハ・ツェンダー干渉計を示す。図 3.10 と違って見えるが、同じものである。

MITOCW Quantum Physics I Part 1: Basic Concepts, Prof. Barton Zwiebach

L2.3 Mach-Zehnder interferometers and beam splitters

<https://ocw.mit.edu/courses/8-04-quantum-physics-i-spring-2016/resources/mach-zehnder-interferometers-and-beam-splitters/>

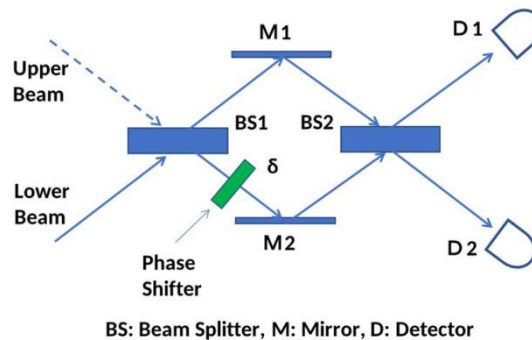


図 3.13 マッハ・ツェンダー干渉計

マッハ・ツェンダー干渉計は、1Q ビットの入力：

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \alpha^2 + \beta^2 = 1 \quad (3.22)$$

に対して、1 量子ビットの出力を与えるものである。ビームスプリッターBS とミラーM は、演算子である。ミラーは光子の方向を変えるだけで何もしない恒等演算子である。

図 3.14 に BS の基底に対する入出力を示す。

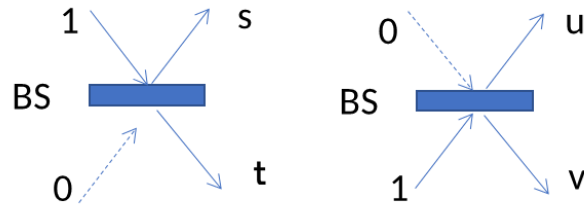


図 3.14 BS の基底に対する入出力

基底が入力した場合の出力が，次式で与えられるとする：

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} s \\ t \end{pmatrix}, s^2 + t^2 = 1 \quad (3.23)$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}, u^2 + v^2 = 1 \quad (3.24)$$

従って，図 3.15 のような入出力に対して

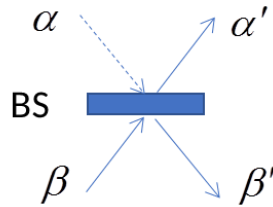


図 3.15 BS の入出力

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \alpha \begin{pmatrix} s \\ t \end{pmatrix} + \beta \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha s + \beta u \\ \alpha t + \beta v \end{pmatrix} = \begin{pmatrix} s & u \\ t & v \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (3.25)$$

を得る．

Balanced Beam Splitter の場合には

$$s^2 = t^2 = u^2 = v^2 = \frac{1}{2} \quad (3.26)$$

となる．従って

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \quad (3.27)$$

が BS の演算子として可能に思えるが

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \dots 1^2 + 1^2 = 2 \dots \text{NG} \quad (3.28)$$

となり，これは確率の規則に反するので採用できない．以下のものは

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \quad (3.29)$$

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \rightarrow \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix}, \quad \frac{1}{2} |\alpha + \beta|^2 + \frac{1}{2} |\alpha - \beta|^2 = |\alpha|^2 + |\beta|^2 = 1 \dots \text{OK} \quad (3.30)$$

であるので可能である。この演算子はアダマールゲートと呼ばれる。図 3.10 と図 3.11 の実験に使われた BS は、この演算子と考えられる。

光路にフェーズシフターの入らない図 3.16 のような場合を考えてみよう。

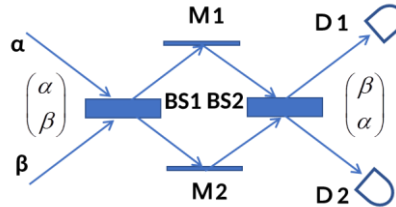


図 3.16 マッハ・ツェンダー干渉計（パウリ X ゲート）

BS1 と BS2 を

$$(BS1) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (BS2) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad (3.31)$$

とすると

$$(BS2)(BS1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{output} = (BS2)(BS1) \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \quad (3.32)$$

であるので、パウリの X ゲート（スワップゲート）になることが分かる。

3.4 重ね合わせと絡み合い

図 3.17 を用いて、1 光子の思考実験を行ってみよう。光子はこれ以上分割できないので、どちらかの検出器で検出される。X 偏光の場合には確率 1 で D1 で検出される。Y 偏光の場合には確率 1 で D2 に検出される。

θ 方向に偏光した連続光 E_0 の実験を行うと x 成分は $E_0 \cos \theta$ 、 y 成分は $E_0 \sin \theta$ となる。明度は光子数に比例するので、光子の個数は $\cos^2 \theta$ と $\sin^2 \theta$ に比例することになる。

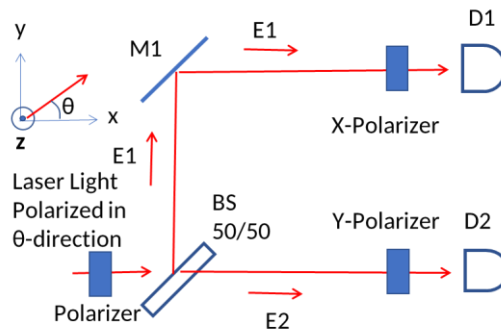


図 3.17 1 光子の思考実験と重ね合わせ

（偏光は光子の進行方向を z 、水平に x 、垂直に y とする。）

θ 方向に偏光した多数の単一光子の実験を行うと、連続光は多数の光子の集団であるから、光子数が十分に多ければ、連続光の明度は光子数に比例するので、単一光子の検出確率になるはずである。このことが正しいければ、ボルンの確率解釈が正しいことになる。図 3.11 の (b) 図の実験が図 3.17 の実験に相当する。図 3.11 の (b) 図の実験結果が $\theta = 45^\circ$ のものであれば、ボルンの確率解釈が妥当と言うことになる。

従って、1 光子の状態を以下のような重ね合わせと仮定することは実験に裏付けられた根拠のあることである：

$$|\psi\rangle = \alpha|X\rangle + \beta|Y\rangle, \quad \alpha^2 + \beta^2 = 1 \quad (3.33)$$

次に、量子現象の際立つ特徴である絡み合い（エンタングルメント）について考察する。図 3.18 は 2Q ビットの入力に 2Q ビットの出力をする簡単な量子回路である。この図に示されるのは、入力では制御ビットと標的ビットは独立（ばらばらに書ける）であるのに対して、出力では制御ビットと標的ビットを別々に書けない（独立でない／絡み合っている）状態になっていることである。図中の \oplus は制御ゲートと言われるもので、標的ビットは制御ビットが $|0\rangle$ のときは素通りし、制御ビットが $|1\rangle$ のときには標的ビットの $|0\rangle$ を $|1\rangle$ に、 $|1\rangle$ を $|0\rangle$ にするものである。ゲートに関しては次節で述べる。

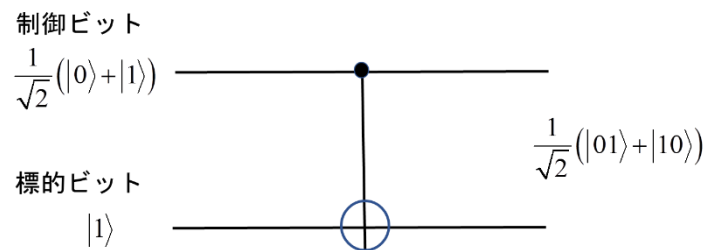


図 3.18 絡み合いを造り出す量子回路

状態 $|01\rangle$ と $|10\rangle$ の観測確率がともに 0.5 で第 1 ビットが 0 なら第 2 ビットは 1, 第 1 ビットが 1 なら第 2 ビットは 0 と観測される。絡み合い状態は、第 1 ビットと第 2 ビットがどんなに離れても解消されない。こんな遠隔作用は古典力学ではあり得ない。

アインシュタインは、この点に異を唱え、何か足りないものがあると主張した。足りないものを補えば、このようなことは起こり得ないと主張した（EPR 論文）。彼は量子の世界が非決定論であることにも反対した。隠れた情報を補えば、確率的な重ね合わせでないはずと主張した。

ベルの理論とそれに基づくアスペらの実験により、彼の考えは否定された。光子の状態が重ね合わせになる事実は、実験に裏付けられるが、何故そうなるのか、その理由は不明といえよう。

3.5 量子計算

量子計算は

(1) 量子ゲート問題

(2) 量子アニーリング問題

に大別される。

量子ゲートと言うのは、古典計算機の論理回路のようなもので、量子ゲートを組み合わせて汎用計算機を作るのが目標である。2030 年頃に 100 万 qubit のものができると言われているが、その道のりは平坦ではない。

量子ビットの状態が一定時間安定して保持される性質をコヒーレンスと言うが、現状では数ナノ秒と非常に短い。量子ビットは、量子力学の原理に基づく複雑な状態を取ることができる。この状態は、量子重ね合わせ状態と呼ばれる。しかし、量子ビットは非常に脆弱で、周囲の環境との相互作用によって量子ビットの状態が崩れ、重ね合わせ状態が失われてしまう。今後の研究によって、より長時間のコヒーレンスを実現することが期待されている。

量子アニーリングは、量子コンピュータを使った最適化問題を解く手法の一つである。最適化問題とは、ある条件下で最大または最小の値を取る変数の組み合わせを求める問題のことである。アニーリングとは、焼き鈍しという熱処理により金属の残留応力を取り除く方法である。

量子アニーリングでは、問題をハミルトニアンという数学的な形式で表現し、それを量子コンピュータ上で実行する。ハミルトニアンは、物理学や量子力学の分野で用いられる概念で、系のエネルギーを表す。最適化問題をハミルトニアンに変換することで、量子コンピュータを使ってその最小エネルギー状態を求めることができる。

ただし、元々のハミルトニアンだけでは、最小エネルギー状態を実現することはできない。ハミルトニアンに時間とともに大から小に変化する揺さぶりを掛けて、元々のハミルトニアンに移行させる。このプロセスが金属を高温から低温に移行させて残留応力の無い状態（最小エネルギー状態）に持って行くアニーリングに似ているので、この名がある。

具体的には、量子ビットを用いたアニーリングマシンに問題を入力し、初期状態を用意して量子アニーリングを実行する。量子アニーリングでは、量子ビットの状態を時間とともに変化させることで、最小エネルギー状態に収束するように制御する。この際、量子ビットの状態を表現する量子重ね合わせ状態を利用することで超並列計算を実現し、古典的なアルゴリズムに比べて高速に最小エネルギー状態を求めることができる。Googleの研究によれば、古典計算機では1万年掛かる問題が200秒で解けたとのことである。

量子アニーリングは、量子コンピュータを使った最適化問題の解法として期待されている。現在、D-Wave Systems社が商用の量子アニーリングマシンを提供している。しかし、D-Wave Systems社の量子アニーラーは、その適用が「2値整数変数の2次関数の最小値問題」に限られている。適用範囲拡張の研究が進んで、任意の最小値問題が高速で解けることが期待されている。

量子ゲート問題と量子アニーリング問題の相互の関係について考えて見よう。量子ゲート問題と量子アニーリング問題は、共に量子コンピュータを使って演算する方法であるが、異なるアプローチを取っている。

量子ゲート問題では、量子ビットに対して複数の量子ゲート操作を施し、最終的に得られた状態を読み取って問題を解く。一方、量子アニーリングでは、問題をハミルトニアンの形式で表現し、最小エネルギー状態を求めることで問題を解決する。

量子ゲート問題と量子アニーリング問題は、一般的には異なる問題に対して効果的な手法であるとされているが、量子ゲート問題でも、ハミルトニアンの形式に変換することができる問題がある。これらの問題に対しては、量子ゲート問題と量子アニーリング問題の両方が適用可能であり、どちらの手法がより効果的かは問題によって異なるであろう。

量子ゲート問題と量子アニーリング問題は相互に関連し、どちらの手法がより適しているかは問題によって異なる。現在、量子コンピュータの性能向上に伴い、より大規模な問題を解決するために、両方の手法を組み合わせたハイブリッドアプローチが注目されている。

量子計算が進歩して、量子計算と AI（人工知能）との融合とか、現在では長時間を要する大規模な数値シミュレーションが、信じがたいほどの短時間で実行される日もさほど遠くないであろう。

第 3 章の参考文献

- [3-1] 宮野健次郎, 古澤 明, 量子コンピューター入門, 日本評論社 (2008).
- [3-2] 中山 茂, 量子アルゴリズム, 技報堂出版 (2014).

第4章 モード展開されたシュレディンガー方程式

前章までで、量子力学およびシュレディンガー方程式に関して、いろんな角度から説明してきた。次章以下で中心になるのは、モード展開されたシュレディンガー方程式であるので、古典力学の振動運動と対比しつつ解説したい。この問題は第2.14節、第2.15節でも触れたが、本章では古典力学との対比を通して、古典力学と量子力学が数学的には同じような枠組みに入ること示そう。簡単のため、座標を x とする1次元空間で考えよう。ここでいう古典力学の振動運動とは弦の運動を意味する。

古典力学の典型的な問題である弦の振動／波動と、量子力学の1量子の振動／波動とには数学的に密接な類似がある。但し、弦の問題では弦を構成する質量の運動すなわち変位分布 $\eta(x, t)$ が対象であるのに対して、1量子の問題の対象は1次元空間における量子の波動関数 $\psi(x, t)$ （複素数値を取り、その絶対値の自乗が存在確率密度分布）である。

結晶格子上的の磁気スピン全体は、イジングモデルと呼ばれる。イジングモデルを形成する個々の磁気スピンは、上向きスピンと下向きスピンという2個の基底よりなる2次元空間であるが、イジングモデルのような多数の磁気スピンの集まったものは、磁気スピンの数を N_q とすると、 2^{N_q} 次元空間になる。このような多次元空間は次章で述べられる。

4.1 古典力学における弦の運動

4.1.1 非加算無限自由度（無限長、分布質量）の弦の波動

無限領域 $-\infty < x < \infty$ における弦の運動は

$$\rho \frac{\partial^2 \eta(x, t)}{\partial t^2} = T \frac{\partial^2 \eta(x, t)}{\partial x^2} \quad (4.1)$$

で与えられる。ここで、 $\eta(x, t)$ は弦の変位、 t は時間、 ρ は質量の線密度、 T は弦の張力である。

変数分離で

$$\eta(x, t) = Y(x)e^{-i\omega t} \quad (4.2)$$

とすると、固有値問題：

$$\frac{d^2 Y(x)}{dx^2} + \frac{\rho \omega^2}{T} Y(x) = 0 \quad (4.3)$$

を得る。有限領域の場合には領域両端で課される境界条件のため ω は離散値となるが、無限領域の場合には任意の実数値を取り得る。波数を k として、モード関数を

$$Y(x, k) = \frac{1}{\sqrt{2\pi}} e^{ikx} \quad (4.4)$$

とすると、式(4.3)より分散方程式（波長と振動数の関係式）：

$$k^2 = \frac{\rho}{T} \omega^2 = \frac{1}{a^2} \omega^2 \quad (4.5)$$

が求まる。ここで、 a は

$$a = \sqrt{\frac{T}{\rho}} \quad (4.6)$$

で、進行波の速度である。波数 k および円周波数 ω は連続実数なので、運動は非加算無限自由度である。

モード関数 $Y(x, k)$ には、次式の直交性がある：

$$\int_{-\infty}^{\infty} Y(x, k) Y(x, k') dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} e^{ik'x} dx = \delta(k + k') \quad (4.7)$$

従って、弦の変位 $\eta(x, t)$ はモードで展開して

$$\eta(x, t) = \int_{-\infty}^{\infty} c(t, k) Y(x, k) dk = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} c(t, k) e^{ikx} dk \quad (4.8)$$

とすると

$$\int_{-\infty}^{\infty} \eta(x, t) Y(x, k') dx = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(t, k) Y(x, k) Y(x, k') dk dx = \int_{-\infty}^{\infty} c(t, k) \delta(k + k') dk = c(t, -k') \quad (4.9)$$

となるので

$$c(t, k) = \int_{-\infty}^{\infty} \eta(x, t) Y(x, -k) dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \eta(x, t) e^{-ikx} dx \quad (4.10)$$

が求まる。これは式 (4.8) のフーリエ逆変換に他ならない。

初期値を

$$\eta(x, t) = u(x), \quad \eta_t(x, t) = v(x) \quad (4.11)$$

とすると、弦の方程式 (4.1) には解析解：

$$\eta(x, t) = f(x - at) + g(x + at) = \frac{1}{2} u(x - at) + \frac{1}{2} u(x + at) - \frac{1}{2a} V(x - at) + \frac{1}{2a} V(x + at) \quad (4.12)$$

が存在する。ここで、 a は式 (4.5) で示したように進行波の速度であり

$$V(x) = \int v(x) dx \quad (4.13)$$

とする。式 (4.12) で与えられる $\eta(x, t)$ が、式 (4.11) を満足することは簡単に確かめられよう。

4.1.2 加算無限自由度（有限長、分布質量）の弦の振動

図 4.1 に示されるような両端固定の弦で質量分布が連続的である振動を考えてみよう。

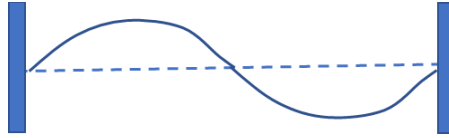


図 4.1 両端固定の弦の振動

弦の長さを L とすると、独立変数の変域が $0 < x < L$ となり、運動方程式は

$$\rho \frac{\partial^2 \eta(x, t)}{\partial t^2} = T \frac{\partial^2 \eta(x, t)}{\partial x^2}, \quad Y(0) = Y(L) = 0 \quad (4.14)$$

で与えられる。

変数分離で

$$\eta(x, t) = Y(x) e^{-i\omega t} \quad (4.15)$$

とすると、固有値問題：

$$\frac{\partial^2 Y}{\partial x^2} = -\frac{\rho}{T} \omega^2 Y, \quad Y(0) = Y(L) = 0 \quad (4.16)$$

を得る。この固有値問題の解として固有値（固有円振動数）と固有関数（振動モード）：

$$\omega_n^2 = \frac{T}{\rho} \left(\frac{(n+1)\pi}{L} \right)^2, \quad n = 0, 1, \dots, \infty \quad (4.17)$$

$$Y_n(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{(n+1)\pi}{L} x\right), \quad n = 0, 1, \dots, \infty \quad (4.18)$$

が求まる。従って、この振動は加算無限個の自由度を有する。式 (4.16) より、モード関数の直交性：

$$(Y_m, Y_n) = \int_0^L Y_m(x) Y_n(x) dx = \delta_{mn} \quad (4.19)$$

が導かれる。この式は式(4.18)を代入して直接確かめても良いが、 $Y_m(x)$ と $Y_n(x)$ が満たす固有値方程式(4.16)の両辺に、 $Y_n(x)$ と $Y_m(x)$ を掛けた後に、辺々相引いて得られる式を、 $0 < x < L$ で積分し部分積分で変形すれば式(4.19)が得られる。

モード関数を使うと、弦の変位 $\eta(x, t)$ は

$$\eta(x, t) = \sum_{n=0}^{\infty} c_n(t) Y_n(x) \quad (4.20)$$

と展開される。ここで

$$c_n(t) = (\eta, Y_n) = \int_0^L \eta(x, t) Y_n(x) dx \quad (4.21)$$

である。式(4.20)を式(4.14)に代入すると、モード毎の振動方程式：

$$\frac{d^2 c_n(t)}{dt^2} = -\omega_n^2 c_n(t) \quad n = 0, 1, \dots, \infty \quad (4.22)$$

が求まる。この微分方程式の解は

$$c_n(t) = A_n \sin \omega_n t + B_n \cos \omega_n t, \quad n = 0, 1, \dots, \infty \quad (4.23)$$

と求まる。

弦の振動の運動エネルギー K ，位置エネルギー V ，全エネルギー H は

$$K = \frac{\rho}{2} \int_0^L \left(\frac{\partial \eta(x, t)}{\partial t} \right)^2 dx = \frac{\rho}{2} \sum_{n=0}^{\infty} \left(\frac{dc_n(t)}{dt} \right)^2 \quad (4.24)$$

$$V = \frac{T}{2} \int_0^L \left(\frac{\partial \eta(x, t)}{\partial x} \right)^2 dx = \frac{\rho}{2} \sum_{n=0}^{\infty} \omega_n^2 c_n^2(t) \quad (4.25)$$

$$H = K + V = \frac{\rho}{2} \sum_{n=0}^{\infty} \left(\frac{dc_n(t)}{dt} \right)^2 + \frac{\rho}{2} \sum_{n=0}^{\infty} \omega_n^2 c_n^2(t) \quad (4.26)$$

と求まる。

4.1.3 有限自由度（有限長，離散質量）の弦の振動

図4.2に示されるような両端固定で質量分布が離散的である弦の振動を考えてみよう。

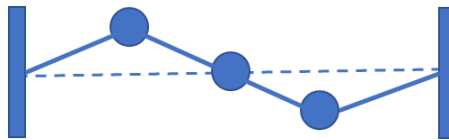


図 4.2 有限自由度の振動（質量が離散分布）

簡単のため，3個の等質量 m の場合を考える。この場合の運動方程式は

$$\begin{pmatrix} d^2 \eta_0(t)/dt^2 \\ d^2 \eta_1(t)/dt^2 \\ d^2 \eta_2(t)/dt^2 \end{pmatrix} = \frac{T}{m\Delta x} \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \eta_0(t) \\ \eta_1(t) \\ \eta_2(t) \end{pmatrix} \quad (4.27)$$

で与えられる。ここで， Δx は質量間の間隔である。

質量の変位を

$$\eta_i(t) = Y_i e^{-i\omega t} \quad (4.28)$$

とすると，固有値問題：

$$\frac{T}{m\Delta x} \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \end{pmatrix} = -\omega^2 \begin{pmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \end{pmatrix} \quad (4.29)$$

を得る.

この固有値問題の解として3個の固有値（固有円振動数）を決める方程式と固有関数（振動モード）:

$$\begin{vmatrix} \frac{m\Delta x}{T}\omega^2 - 2 & 1 & 0 \\ 1 & \frac{m\Delta x}{T}\omega^2 - 2 & 1 \\ 0 & 1 & \frac{m\Delta x}{T}\omega^2 - 2 \end{vmatrix} = 0 \quad (4.30)$$

$$\vec{Y}_n = (Y_{n0} \ Y_{n1} \ Y_{n2})^T, \quad n=0,1,2 \quad (4.31)$$

が求まる. 従って, この振動は有限個の自由度を有する. 式(4.29)より, モード関数の直交性:

$$(\vec{Y}_m, \vec{Y}_n) = \sum_{i=0}^2 Y_{mi} Y_{ni} = \delta_{mn} \quad (4.32)$$

が導かれる. \vec{Y}_m と \vec{Y}_n が満たす固有値方程式(4.29)の両辺に, \vec{Y}_n と \vec{Y}_m を掛けた後に, 辺々相引いて得られる式を, 変形すれば式(4.32)が得られる.

弦の変位 $\eta_i(t)$ は

$$\eta_i(t) = \sum_{n=0}^2 c_n(t) Y_{ni} \quad (4.33)$$

と展開される. ここで

$$c_n(t) = (\eta, Y_n) = \sum_{i=0}^2 \eta_i(t) Y_{ni} \quad (4.34)$$

である. 式(4.33)を式(4.27)に代入すると, モード毎の振動方程式:

$$\frac{d^2 c_n(t)}{dt^2} = -\omega_n^2 c_n(t) \quad n=0,1,2 \quad (4.35)$$

が求まる. この微分方程式の解は

$$c_n(t) = A_n \sin \omega_n t + B_n \cos \omega_n t, \quad n=0,1,2 \quad (4.36)$$

と求まる.

4.2 量子力学における1量子の運動

4.2.1 非加算無限自由度（無限領域, 連続確率密度分布）の1量子の運動

簡単のため, 1次元空間 $-\infty < x < \infty$ における1電子の自由運動のような1量子の運動を考える. 時間を t , 質量を m , 波動関数を $\psi(x,t)$ とすると, シュレディンガーの波動方程式と境界条件は

$$i\hbar \frac{\partial \psi(x,t)}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V \right) \psi(x,t) = \hat{H} \psi(x,t), \quad -\infty < x < \infty, \quad 0 < t < \infty \quad (4.37)$$

$$\psi(\pm\infty, t) = 0, \quad |\psi_x(\pm\infty, t)| < \infty \quad (4.38)$$

と与えられる. ここで, \hbar を換算プランク定数（ディラック定数）, p を運動量, V を位置エネルギーとすると, 全エネルギー H , 運動量演算子 \hat{p} , ハミルトニアン \hat{H} は次式で与えられる:

$$H = \frac{p^2}{2m} + V \quad (4.39)$$

$$\hat{p} = -i\hbar \frac{\partial}{\partial x} \quad (4.40)$$

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V = \frac{\hat{p}^2}{2m} + V \quad (4.41)$$

変数を分離して

$$\psi(x, t) = \Psi(x) e^{-i\omega t} \quad (4.42)$$

を式(4.37)に代入すると、固有値問題：

$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V \right) \Psi(x) = \hbar\omega \Psi(x) \quad (4.43)$$

を得る。式(4.43)に固有関数（モード関数）：

$$\Psi(x, k) = \frac{1}{\sqrt{2\pi}} e^{ikx} \quad (4.44)$$

を代入すると、分散方程式：

$$\hbar\omega = \frac{\hbar^2}{2m} k^2 + V \quad (4.45)$$

を得る。波数 \hbar および円周波数は連続な数値であるので、この波動の自由度は非加算無限である。モード関数 $\Psi(x, k)$ には、直交性：

$$\int_{-\infty}^{\infty} \Psi(x, k) \Psi^*(x, k') dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} e^{-ik'x} dx = \delta(k - k') \quad (4.46)$$

がある。

シュレディンガーの方程式の解は、次式で与えられる：

$$\psi(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \phi(k) e^{ikx} e^{-i\omega(k)t} \quad (4.47)$$

ここで

$$\psi(x, 0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \phi(k) e^{ikx} \quad \text{at } t=0 \rightarrow \phi(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx \psi(x, 0) e^{-ikx} \quad (4.48)$$

とする。

4.2.2 加算無限自由度（有限領域、無限離散確率）の1量子の運動

箱の中 $0 < x < L$ の量子の運動について考えて見よう。波動関数（量子の状態） $\psi(x, t)$ に関する境界値問題は

$$i\hbar \frac{\partial \psi(x, t)}{\partial t} = \hat{H} \psi(x, t), \quad 0 < x < L \quad (4.49)$$

$$\psi(0, t) = \psi(L, t) \quad (4.50)$$

で定義される。ここで、

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V = \hat{p}^2 + V, \quad \hat{p} = -i\hbar \frac{\partial}{\partial x}, \quad H = \frac{p^2}{2m} + V \quad (4.51)$$

である。

変数を分離して

$$\psi(x, t) = \Psi(x) e^{-i\omega t} \quad (4.52)$$

とすると、固有値問題：

$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V \right) \Psi(x) = \hbar\omega \Psi(x) \quad (4.53)$$

$$\Psi(0) = \Psi(L) = 0 \quad (4.54)$$

を得る。 $V=0$ のときには、この固有値問題の解として固有値（固有円振動数）と固有関数（振動モード）：

$$\omega_n = \frac{1}{2m} \left(\frac{(n+1)\pi}{L} \right)^2, \quad n = 0, 1, \dots, \infty \quad (4.55)$$

$$\Psi_n(x) = \sqrt{\frac{2}{L}} \sin \left(\frac{(n+1)\pi}{L} x \right), \quad n = 0, 1, \dots, \infty \quad (4.56)$$

が求まる。従って、この振動は加算無限個の自由度を有する。式(4.53)より、モード関数の直交性：

$$(\Psi_m, \Psi_n) = \int_0^L \Psi_m(x) \Psi_n(x) dx = \delta_{nm} \quad (4.57)$$

が導かれる。この式は式(4.56)を代入して直接確かめても良いが、 $\Psi_m(x)$ と $\Psi_n(x)$ が満たす固有値方程式(4.53)の両辺に、 $\Psi_n(x)$ と $\Psi_m(x)$ を掛けた後に、辺々相引いて得られる式を、 $0 < x < L$ で積分し部分積分で変形すれば式(4.57)が得られる。

モード関数を使うと、波動関数 $\psi(x, t)$ は

$$\psi(x, t) = \sum_{n=0}^{\infty} c_n(t) \Psi_n(x) \quad (4.58)$$

と展開される。ここで

$$c_n(t) = (\psi, \Psi_n) = \int_0^L \psi(x, t) \Psi_n(x) dx \quad (4.59)$$

である。式(4.20)を式(4.14)に代入すると、モード毎の振動方程式：

$$\frac{dc_n(t)}{dt} = -i\omega_n c_n(t), \quad n = 0, 1, \dots, \infty \quad (4.60)$$

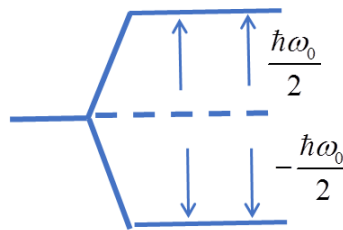
が求まる。この微分方程式の解は

$$c_n(t) = c_n(0) e^{-i\omega_n t}, \quad n = 0, 1, \dots, \infty \quad (4.61)$$

と求まる。

4.2.3 有限自由度（有限領域、有限離散確率）の1量子の運動[4-2]

有限自由度の問題に簡単に触れて見たい。この問題は、すでに第2.15節で2準位原子の問題として触れた。2準位原子と言うのは、図4.3に視されるように、最外殻に1個の電子を持つ原子で、その電子が熱や電磁波により励起されて基底状態からより高いエネルギー状態に移行することができ、エネルギーを失って放射を行って基底状態に戻ることができる。この放射で可視光線が放出されることが多い。



エネルギー準位

図4.3 2準位原子

ディラックの記法を用いて、電子の状態 $|\psi(t)\rangle$ は2個の状態 $|0\rangle$ と状態 $|1\rangle$ の重ね合わせ状態にあり

$$|\psi(t)\rangle = c_0(t)|0\rangle + c_1(t)|1\rangle = \begin{pmatrix} c_0(t) \\ c_1(t) \end{pmatrix} \quad (4.62)$$

と表される。ここで、係数 $c_0(t)$ と $c_1(t)$ は複素数値で

$$|c_0(t)|^2 + |c_1(t)|^2 = 1 \quad (4.63)$$

を満足し、 $|c_0(t)|^2$ と $|c_1(t)|^2$ は観測時に状態 $|0\rangle$ あるいは状態 $|1\rangle$ が出る確率である。

状態 $|\psi(t)\rangle$ は、この場合のシュレディンガー方程式：

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = \hat{H}|\psi(t)\rangle \quad (4.64)$$

に従う。式 (4.62) を代入すると

$$i\hbar \begin{pmatrix} dc_0(t)/dt \\ dc_1(t)/dt \end{pmatrix} = \hat{H} \begin{pmatrix} c_0(t) \\ c_1(t) \end{pmatrix} \quad (4.65)$$

と書ける。ここで

$$\hat{H} = \begin{pmatrix} -\hbar\omega_0/2 & 0 \\ 0 & \hbar\omega_0/2 \end{pmatrix} \quad (4.66)$$

である。

正弦運動を仮定して

$$c_n(t) = c_n e^{-i\omega t} \quad (4.67)$$

とすると、固有値問題：

$$\begin{pmatrix} -\hbar\omega_0/2 & 0 \\ 0 & \hbar\omega_0/2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \hbar\omega \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \quad (4.68)$$

を得る。2 個の振動モードの円周波数は

$$\begin{vmatrix} \hbar(\omega + \omega_0/2) & 0 \\ 0 & \hbar(\omega - \omega_0/2) \end{vmatrix} = 0 \rightarrow \omega = \mp \frac{\omega_0}{2} \quad (4.69)$$

となり、モードは

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (4.70)$$

という当然の結果を得る。

第 4 章の参考文献

[4-1] 宮野健次郎, 古澤 明, 量子コンピューター入門, 日本評論社 (2008).

[4-2] 中山 茂, 量子アルゴリズム, 技報堂出版 (2014).

第5章 量子アニーリング問題

5.1 量子アニーリングの現状

最近、量子情報処理技術は急速に進展しつつある。その影響は多方面に及ぶと予測されており、広範な分野へ大きな影響を及ぼすであろう。

D-Wave 社はこのアニーリング問題に特化した専用量子計算機を開発しており、在来型計算機（古典計算機）では1万年掛かるような組み合わせ爆発問題を200秒で解いてしまうなど、世の中に大きな衝撃を与えている。しかし、D-Wave 社の専用量子計算機でできるのは、イジング問題やクボ問題と呼ばれる2値整数変数の2次関数の最小値問題に限定されている。従って、この量子計算機が万能というわけではない。その理由はこの計算機で解ける問題が、下記のような整数2値変数 x_i , $i = 0, 1, \dots, n-1$ の2次関数の最小値問題：

$$\min_{\mathbf{x} \in \{0,1\}^n} \left(a + \sum_{i=0}^{n-1} b_i x_i + \sum_{i=0}^{n-1} \sum_{j=1}^{n-1} c_{ij} x_i x_j \right)$$

に限られるからである。ここで、 a, b_i, c_{ij} は通常の定数である。要するに解き得る問題が限定されている。もっと一般的な問題に適用されるべきであろう。

本章では、先ず量子アニーリングの理論の解説を行い、その後で、量子ビットによって超並列計算を可能にする量子計算機によって、任意の大域最小値探索を可能にする方法を考察する。

そのために取り組むべき課題は下記のように三つある：

- (1) 10進整数変数の2進整数変数への変換
- (2) 10進連続変数の2進離散変数近似
- (3) 2次関数でない一般の関数への拡張

この課題を解決するためには、それに必要な基本的数理を開発することが求められる。

例えば、数値シミュレーションで多用される大規模連立1次方程式への適用が求められよう。最も重要と思われるのは、大規模ニューラル・ネットワークの学習への適用であろう。短時間で学習できるようになれば、AIに画期的な変革をもたらすであろう。本著書ではこれらの問題についても検討している。

量子アニーリングの技術が成熟して大量のデータ処理が可能になると、現在では想像もできないような大量のデータのモニタリングが可能になるであろう。例えば、陸上の車や空飛ぶ車の管制である。また、それを支える技術として高速・高精度の衛星測位技術の開発が求められよう。そのためには、量子計算による高速計算が必須であろう。

D-Wave 社のアニーリング・マシンの適用範囲を拡大するための基本的数理を考察し、数理の実証も行った。古典計算機で行った検証計算の結果も示す。量子アニーリングとは、各基底のエネルギーの最小値を求める方法である。検証計算とは、そのことを数値計算を実施して実証することである。本来、量子計算機で検証できればよいが、現在の量子計算機の機能を超えるので、古典計算機で行った。

5.2 量子アニーリングの理論

5.2.1 量子アニーリングによる大域最小値探索

量子アニーリングとは、実数列 $E_0, E_1, \dots, E_{N_b-1}$ の大域的 minimum を、量子力学の原理とアニーリング（焼き鈍し）を組み合わせる方法である。アニーリングの元来の意味は、高温にした金属の温度を徐々に下げることにより、残留応力を取り除く方法である。高温から低温に移すことは、残留歪エネルギー

を最小化することを意味する。実数列 $E_0, E_1, \dots, E_{N_b-1}$ が何らかの物理系の基底のエネルギーとし、この物理系は高温では色々な基底が混在しているとし、低温に移行することにより、混在した状態からエネルギー最小の基底に移行するとする。別の言い方をすれば、高温でほぼ等しかった各基底の確率が、低温に移行することにより、エネルギー最小の基底の確率が1になり、他の基底の確率が0になることを想定する。この状態で測定すれば、エネルギー最小の基底が測定されることになる。量子系は正にそのような系で、その時間進化（高温から低温への移行）はシュレディンガーの波動方程式で記述される。

上述のイメージを図 5.1 に示す。時間進化を誘起するために、外部から揺さぶりを掛ける。図 5.1 では高温から低温への移行の代わりに、大から小に移行する機械的な揺さぶりを与える。凸凹の地形の上に豆粒をばらまき、箱を揺する。最初は大きく、次第に小さく揺さぶると、豆粒が段々一番低い所に集まる。重力の作用がシュレディンガー方程式に、豆粒が多数あることが量子計算の並列計算（重ね合わせ）に、揺らし方がアニーリングに相当する。スパコンの並列計算では、多数の演算器を用いて、多数の豆粒の処理を行う。いわば、多数の古典計算機を同時に動かすことに相当する。量子計算では一つの量子演算器が量子の重ね合わせ原理に基づく超並列計算を実施する点で、古典計算とは完全に異なる。

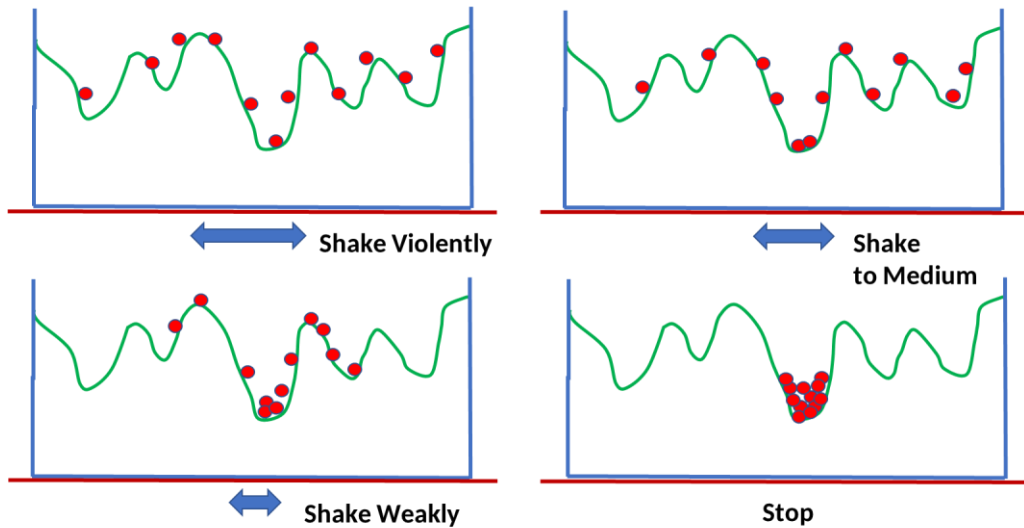


図 5.1 Image of forced oscillation to obtain global minimum.

5.2.2 量子アニーリングの理論

まず、量子力学の原理を簡単に説明してから、量子アニーリングを説明する。2 値 0 と 1 を取る 1 個の量子からなる系の状態を

$$|\psi\rangle_1 = c_0|0\rangle + c_1|1\rangle, \quad (5.1)$$

$$|c_0|^2 + |c_1|^2 = 1 \quad (5.2)$$

とする。ここで、 $|0\rangle$ と $|1\rangle$ は基底で、係数 c_0 と c_1 はその複素数値を取る係数である。測定時に $|0\rangle$ が観測される確率は $|c_0|^2$ で、 $|1\rangle$ が観測される確率は $|c_1|^2$ である。この系の状態は、次のような列ベクトルと考えても良い：

$$|\psi\rangle_1 = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = c_0|0\rangle + c_1|1\rangle \quad (5.3)$$

すなわち、 $|\psi\rangle_1$ は $|0\rangle$ と $|1\rangle$ を基底とする 2 次元空間である。

2 値を取る N_q 個の量子ビット (qubit) が作る系の状態を $|\psi\rangle_{N_q}$ とする. 物理的には, 結晶格子上的の磁気スピンの数学モデルになっていて, イジングモデルと呼ばれる. イジングモデルの場合には 2 値状態は上向きスピン \uparrow (+1) と下向きスピン \downarrow (-1) である. 簡単のために $N_q = 2$ の場合について考えると

$$\begin{aligned} |\psi\rangle_2 &= (c_0|0\rangle + c_1|1\rangle) \otimes (c_2|0\rangle + c_3|1\rangle) = c_0c_2|0\rangle \otimes |0\rangle + c_0c_3|0\rangle \otimes |1\rangle + c_1c_2|1\rangle \otimes |0\rangle + c_1c_3|1\rangle \otimes |1\rangle \\ &= C_0|00\rangle + C_1|01\rangle + C_2|10\rangle + C_3|11\rangle \end{aligned} \quad (5.4)$$

と書ける. ここで

$$C_0 = c_0c_2, \quad C_1 = c_0c_3, \quad C_2 = c_1c_2, \quad C_3 = c_1c_3 \quad (5.5)$$

$$|00\rangle = |0\rangle \otimes |0\rangle, \quad |01\rangle = |0\rangle \otimes |1\rangle, \quad |10\rangle = |1\rangle \otimes |0\rangle, \quad |11\rangle = |1\rangle \otimes |1\rangle \quad (5.6)$$

とした. 改めて

$$C_0 \rightarrow c_0, \quad C_1 \rightarrow c_1, \quad C_2 \rightarrow c_2, \quad C_3 \rightarrow c_3 \quad (5.7)$$

$$|00\rangle \rightarrow |0\rangle, \quad |01\rangle \rightarrow |1\rangle, \quad |10\rangle \rightarrow |2\rangle, \quad |11\rangle \rightarrow |3\rangle \quad (5.8)$$

と書くことにすると

$$\begin{aligned} |\psi\rangle_2 &= c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle \\ &= c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + c_3|3\rangle \end{aligned} \quad (5.9)$$

と書ける. ここで

$$\begin{aligned} |0\rangle = |00\rangle = |0\rangle \otimes |0\rangle &= \begin{pmatrix} 1 \times 1 \\ 0 \\ 0 \times 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle = |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \times 0 \\ 1 \\ 0 \times 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\ |2\rangle = |10\rangle = |1\rangle \otimes |0\rangle &= \begin{pmatrix} 0 \times 1 \\ 0 \\ 1 \\ 1 \times 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |3\rangle = |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \times 0 \\ 1 \\ 0 \\ 1 \times 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned} \quad (5.10)$$

であり, $|\psi\rangle_2$ は $|0\rangle, |1\rangle, |2\rangle, |3\rangle$ を基底とする $4 (= 2^2)$ 次元空間である.

同様に考えると, $|\psi\rangle_{N_q}$ は $|0\rangle, |1\rangle, \dots, |2^{N_q} - 1\rangle$ を基底:

$$\begin{aligned} |0\rangle &= |00 \dots 00\rangle = |0\rangle \otimes |0\rangle \otimes \dots \otimes |0\rangle \otimes |0\rangle = (1, 0, 0, 0, 0, \dots, 0, 0)^T \\ |1\rangle &= |00 \dots 01\rangle = |0\rangle \otimes |1\rangle \otimes \dots \otimes |0\rangle \otimes |0\rangle = (0, 1, 0, 0, 0, \dots, 0, 0)^T \\ &\vdots \\ |2^{N_q} - 2\rangle &= |11 \dots 10\rangle = |1\rangle \otimes |1\rangle \otimes \dots \otimes |1\rangle \otimes |0\rangle = (0, 0, 0, 0, 0, \dots, 1, 0) \\ |2^{N_q} - 1\rangle &= |11 \dots 11\rangle = |1\rangle \otimes |1\rangle \otimes \dots \otimes |1\rangle \otimes |1\rangle = (0, 0, 0, 0, 0, \dots, 0, 1) \end{aligned} \quad (5.11)$$

とする N_b 次元空間である. ここで

$$N_b = 2^{N_q} \quad (5.12)$$

である. すなわち

$$|\psi\rangle_{N_q} = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + \dots + c_{N_b-2}|N_b-2\rangle + c_{N_b-1}|N_b-1\rangle \quad (5.13)$$

と書ける.

量子アニーリングとは, 基底エネルギーの数列 $E_0, E_1, \dots, E_{N_b-1}$ の大域的最小値を, アニーリング効果を取り入れたシュレディンガー方程式を時間進化させて, 大域的最小値に到達する方法である.

状態 $|\psi\rangle_{N_q}$ 各基底の係数 $c_0, c_1, \dots, c_{N_b-1}$ の時間進化は, 第 4 章で説明したように

$$i\hbar \begin{pmatrix} dc_0/dt \\ dc_1/dt \\ \vdots \\ dc_{N_b-1}/dt \end{pmatrix} = \hat{H}(t) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N_b-1} \end{pmatrix} \quad (5.14)$$

で与えられる。ここで

$$\begin{aligned} \hat{H}(t) &= A(t)\Gamma\hat{H}_A(t) + B(t)\hat{H}_B \\ A(t) &= 1 - \frac{t}{\tau} \quad B(t) = \frac{t}{\tau} \\ \hat{H}(0) &= \hat{H}_A \quad \hat{H}(\tau) = \hat{H}_B \end{aligned} \quad (5.15)$$

とする[5-5, p. 211]. ハミルトニアン \hat{H}_A と \hat{H}_B は, 例えば 3Q ビットの場合には

$$\hat{H}_A = \Gamma \begin{pmatrix} 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \end{pmatrix}, \quad \hat{H}_B = \begin{pmatrix} E_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & E_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & E_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & E_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & E_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_7 \end{pmatrix} \quad (5.16)$$

で与えられる。

$\hat{H}_A = 0$ で \hat{H}_B が時間の関数でない場合には, 時間進化の微分方程式は簡単に解けて, その解は

$$c_n(t) = c_n(0) \exp\left(-i \frac{E_n}{\hbar} t\right), \quad n = 0, 1, \dots, N_b - 1 \quad (5.17)$$

と求まる。従って

$$|c_n(t)|^2 = |c_n(0)|^2, \quad n = 0, 1, \dots, N_b - 1 \quad (5.18)$$

であるので, 時間進化が起こらないことになってしまう。

時間進化を起こらせるためには揺さぶりを掛けないといけない。その役割を果たすのが \hat{H}_A である。

\hat{H}_A はその目的に合うようにパウリのスワップ行列 (X 行列) :

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (5.19)$$

を元にして作られている。最初は強く, 次第に弱くなるような揺さぶりを掛けるとエネルギー最小の基底の確率が 1 となり, 他の基底の確率が 0 になるように時間進化する。この状態で測定すれば, エネルギー最小の基底が観測される。このプロセスは, 金属の残留応力を除去するのに使われる焼きなまし (アニーリング) に似ているので, 量子アニーリングと呼ばれる。焼きなましは, ひずみエネルギーを最小にするプロセスである。

\hat{H}_A の作り方を述べる。 $N=1$ の場合は, パウリの X 行列 (スワップ行列) の符号を - にしたもの, $N=2$ の場合はこれを左右反転して右上に, こうしてできた上 2 行を上下反転して下 2 行にはめたものである。同じ手続きで, $N=n$ から $N=n+1$ を作る。C 言語のコードを Code 1 に示す :

Code 1 Codes for generating \hat{H}_A in C language

```
void GenOfHA(int Nq) {
    int i, j;
```

```

// upper left
HA[0][0] = 0.0;
for (j = 1; j < Nb/2; j++)
    HA[0][j] = Rvs(HA[0][j-1]);

for (i = 1; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[i][j] = Rvs(HA[i-1][j]);

// lower left
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[Nb/2+i][j] = Rvs(HA[i][j]);

// upper right
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[i][Nb/2+j] = HA[Nb/2+i][j];

// lower right
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[Nb/2+i][Nb/2+j] = HA[i][j];
}

double Rvs(double x) {
    if (x == 0.0)
        return -1.0;
    else
        return 0.0;
}

```

数値計算時の時間進行は、式 (5. 14) で与えられる常微分方程式を直接解いても良いし、式 (5. 14) から導かれる t_{i-1} から t_i への時間進行演算子：

$$\exp\left(-i\frac{\hat{H}(t_{i-1})}{\hbar}\Delta t\right) \quad (5. 20)$$

を用いて

$$\vec{c}(t_i) = \exp\left(-i\frac{\hat{H}(t_{i-1})}{\hbar}\Delta t\right)\vec{c}(t_{i-1}) \quad (5. 21)$$

としてもよい。ただし、時間ステップ Δt を十分小さく取ることが肝要である。

図 5. 2 に重ね合わせに基づく超並列アニーリング計算のフローを示す。D-Wave に倣って、各アニーリングの最後に確率 $|c_n|^2$ を測定する。アニーリングの結果、確率 $|c_n|^2$ の高いものが測定される。D-Wave に倣って、アニーリングを 1000 回繰り返す、最小値候補の頻度分布を取り最小値を決定する。ハミルトニアン \hat{H}_A と \hat{H}_B の計算も重ね合わせの一部として行われる。事前に \hat{H}_B を求めるとしたら、エネルギー E_n , $n=0,1,\dots,N_b$ の最小値を量子アニーリングで求める価値が半減する。その理由は大規模な系 (N_b が大きい) の場合には、 \hat{H}_B を求めること自体が時間が掛かるし、最小値を求めるのに量子アニーリング以外の方法も可能だからである。

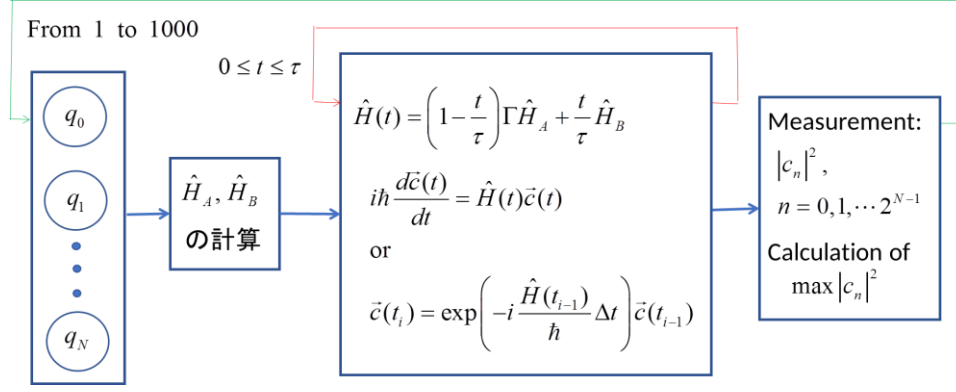


図 5.2 量子アニーリング (Quantum Annealing)

5.3. 量子アニーリングの計算例

第 5.1 節でその必要性を述べ、第 2 節でその理論を述べた拡張された量子アニーリングを適用した簡単な計算例を以下に示す。

5.3.1 一変数の場合の量子アニーリング大域最小値探索

(1) 3Q ビット, 1 整数変数, 2 次関数の最小値問題

通常の整数変数の最小値問題を、量子アニーリングで解く。 p を通常の整数として、目的関数 $f(p)$ を

$$f(p) = (p - a)^2 \quad (5.22)$$

とする 1 整数変数の最小値問題を考える。 p を 2 進数で表すと

$$\begin{aligned} p &= \sum_{\rho=0}^2 2^{\rho} q_{\rho} = q_0 + 2q_1 + 4q_2, \\ p^2 &= \sum_{\rho=0}^2 2^{\rho} q_{\rho} \sum_{\sigma=0}^2 2^{\sigma} q_{\sigma} = \sum_{\rho=0}^2 \sum_{\sigma=0}^2 2^{\rho+\sigma} q_{\rho} q_{\sigma} = q_0^2 + 4q_1^2 + 16q_2^2 + 4q_0q_1 + 16q_1q_2 + 8q_2q_0 \\ &= q_0 + 4q_1 + 16q_2 + 4q_0q_1 + 16q_1q_2 + 8q_2q_0 \end{aligned} \quad (5.23)$$

となる。 3Q ビットの場合には p の範囲は

$$p = 0, 1, 2, 3, 4, 5, 6, 7 \quad (5.24)$$

である。式 (5.23) では、クボ変数の性質：

$$q_i^2 = q_i, \quad i = 0, 1, 2 \quad (5.25)$$

を用いているが、用いなくともよい。式 (5.23) を用いると

$$f(p) = (p - a)^2 = q_0 + 4q_1 + 16q_2 + 4q_0q_1 + 16q_1q_2 + 8q_2q_0 \quad (5.26)$$

を得る。

式 (5.26) より

$$\begin{aligned} E_0 &= f_p(0) = E_{000} = f_q(0, 0, 0) = a^2, & E_4 &= f_p(4) = E_{100} = f_q(1, 0, 0) = 16 - 8a + a^2, \\ E_1 &= f_p(1) = E_{001} = f_q(0, 0, 1) = 1 - 2a + a^2, & E_5 &= f_p(5) = E_{101} = f_q(1, 0, 1) = 25 - 10a + a^2, \\ E_2 &= f_p(2) = E_{010} = f_q(0, 1, 0) = 4 - 4a + a^2, & E_6 &= f_p(6) = E_{110} = f_q(1, 1, 0) = 36 - 12a + a^2, \\ E_3 &= f_p(3) = E_{011} = f_q(0, 1, 1) = 9 - 6a + a^2, & E_7 &= f_p(7) = E_{111} = f_q(1, 1, 1) = 49 - 14a + a^2 \end{aligned} \quad (5.27)$$

となる。

各基底の係数 $c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7$ の時間進化は

$$i\hbar \begin{pmatrix} dc_0/dt \\ dc_1/dt \\ \vdots \\ dc_{2^N-1}/dt \end{pmatrix} = \hat{H}(t) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^N-1} \end{pmatrix} \quad (5.28)$$

で与えられる。ここで

$$\begin{aligned} \hat{H}(t) &= A(t)\Gamma\hat{H}_A(t) + B(t)\hat{H}_B \\ A(t) &= 1 - \frac{t}{\tau} \quad B(t) = \frac{t}{\tau} \\ \hat{H}(0) &= \hat{H}_A \quad \hat{H}(\tau) = \hat{H}_B \end{aligned} \quad (5.29)$$

とする。 $\hat{H}_A(t)$ と $\hat{H}_B(t)$ は、下記で示したものであり

$$\hat{H}_A = \Gamma \begin{pmatrix} 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \end{pmatrix}, \hat{H}_B = \begin{pmatrix} E_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & E_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & E_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & E_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & E_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_7 \end{pmatrix} \quad (5.30)$$

で与えられる。

初期条件を

$$|\psi\rangle_{t=0} = \frac{1}{\sqrt{2^3}} (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T \quad (5.31)$$

として、式(5.28)で与えられる常微分方程式をオイラー法で解いた。以下のパラメーターを計算に用いた：

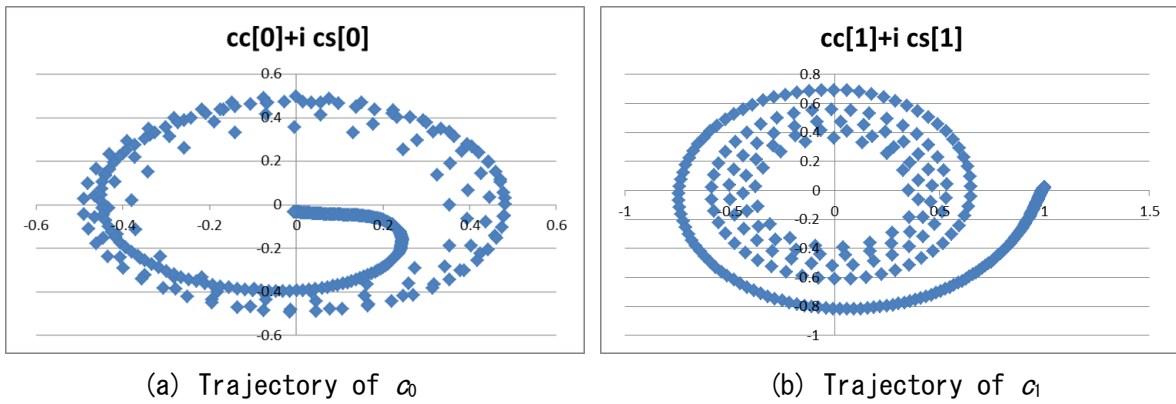
$$a=1.0, \quad \hbar=0.1, \quad \Gamma=1.0, \quad dt=0.00001, \quad \tau=3.0 \quad (5.32)$$

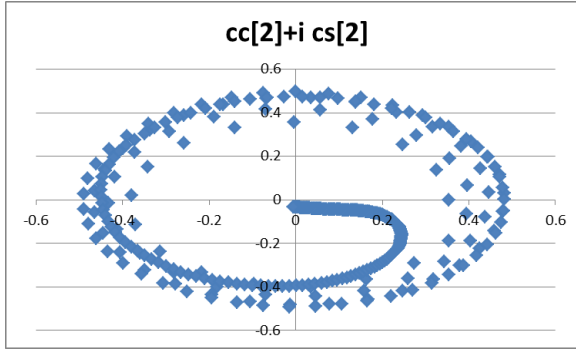
計算結果を図 5.3 に示す。 $p=1$ すなわち基底 $(0,0,1)$ で $|c_1|^2=1$ になっている。正確には

$$\text{Re}[c_1]=1.00181, \quad \text{Im}[c_1]=0.01847, \quad p_2=|c_2|^2=1.00198 \quad (5.33)$$

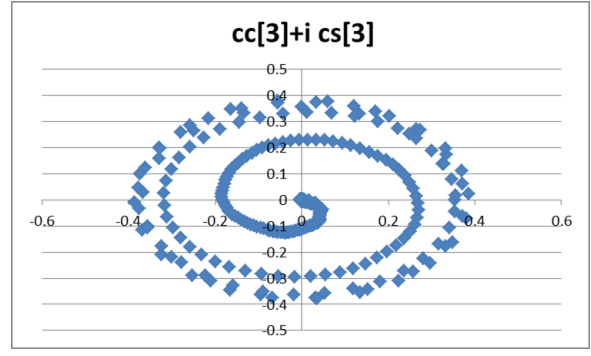
が得られた。

計算に用いられた C 言語で書かれた計算コードを付録 A に示す。

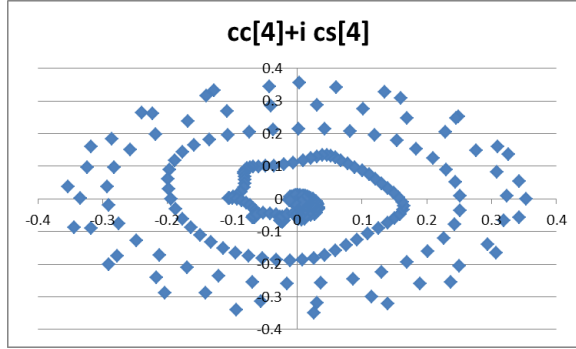




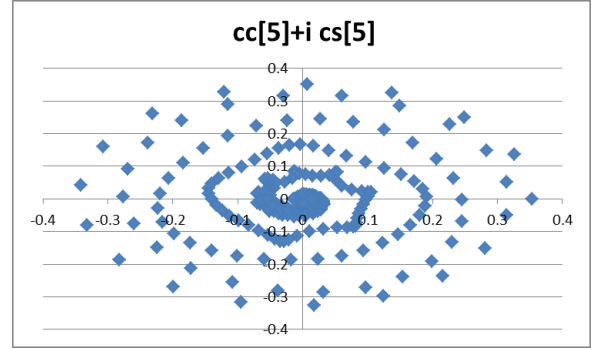
(c) Trajectory of c_2



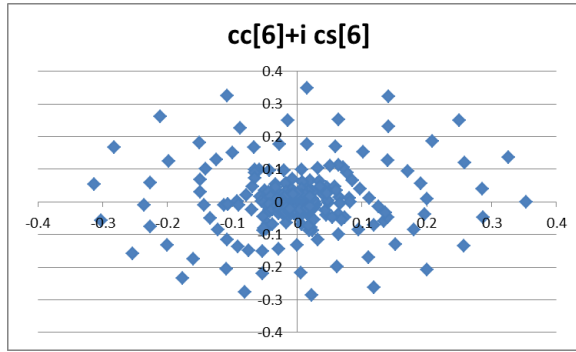
(d) Trajectory of c_3



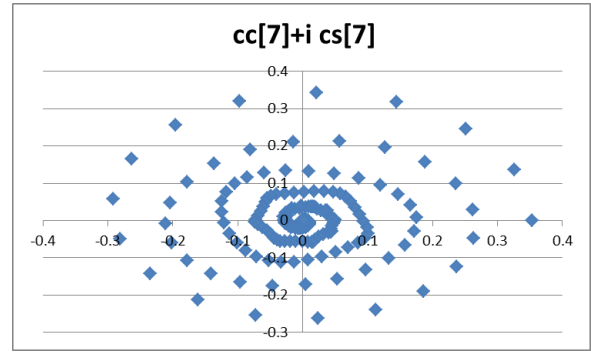
(e) Trajectory of c_4



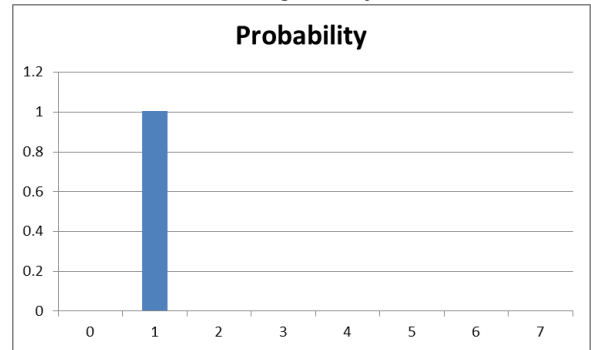
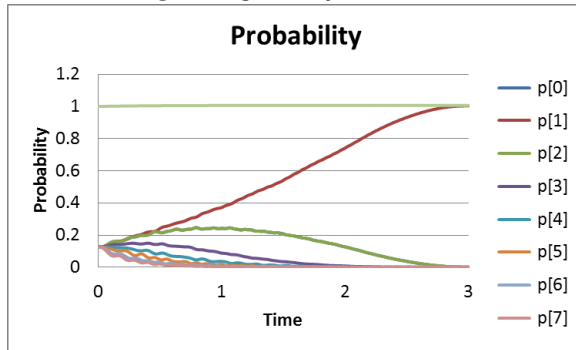
(f) Trajectory of c_5



(g) Trajectory of c_6



(h) Trajectory of c_7



Probability of base 0, 1, 2, 3, 4, 5, 6 and 7

図 5.3 Solution of $\min_{p=0,1,\dots,7} (p-a)^2$ by quantum annealing ($N=3$; $p=0,1,2,3,4,5,6,7$)

(2) 3Q ビット, 1 連続変数, 2 次関数の最小値問題

(1) の最小値問題は 1 整数変数 p の最小値問題であったが, 1 連続変数 x の最小値問題に拡張する.

また, 目的関数を 2 次関数に限定しないで, 一般の関数に拡張する. すなわち

$$f(x) = (x-a)^2 \quad (5.34)$$

とする。このケースでは2次関数としたが、以下の(3)では2次関数でない。まず、連続変数 x を離散化して

$$x_n = -L + n\Delta x, \quad n = 0, 1, \dots, 2^N - 1 \quad (5.35)$$

とする。次に、整数 i を2進整数で置き換える：

$$n = \sum_{i=0}^{2^N-1} 2^i q_i = 2^0 q_0 + 2^1 q_1 + \dots + (2^N - 1) q_{2^N-1} \quad (5.36)$$

式(5.36)に式(5.35)を代入すると

$$x = x(q_0, q_1, \dots, q_{2^N-1}) = -L + \Delta x \sum_{i=0}^{2^N-1} 2^i q_i, \quad n = 0, 1, \dots, 2^N - 1 \quad (5.37)$$

を得る。これを式(5.34)に代入すると

$$f(x) = (x-a)^2 = \left(-L - a + \Delta x \sum_{i=0}^{2^N-1} 2^i q_i \right)^2 = (L-a)^2 - 2(L-a)\Delta x \sum_{i=0}^{2^N-1} 2^i q_i + \Delta x^2 \sum_{i=0}^{2^N-1} \sum_{j=0}^{2^N-1} 2^{i+j} q_i q_j \quad (5.38)$$

式(5.38)より

$$\begin{aligned} E_0 &= f_x(x_0) = E_{000} = f_q(0, 0, 0), & E_4 &= f_x(x_4) = E_{100} = f_q(1, 0, 0), \\ E_1 &= f_x(x_1) = E_{001} = f_q(0, 0, 1), & E_5 &= f_x(x_5) = E_{101} = f_q(1, 0, 1), \\ E_2 &= f_x(x_2) = E_{010} = f_q(0, 1, 0), & E_6 &= f_x(x_6) = E_{110} = f_q(1, 1, 0), \\ E_3 &= f_x(x_3) = E_{011} = f_q(0, 1, 1), & E_7 &= f_x(x_7) = E_{111} = f_q(1, 1, 1) \end{aligned} \quad (5.39)$$

となる。

各基底の係数 $c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7$ の時間進化は

$$i\hbar \begin{pmatrix} dc_0/dt \\ dc_1/dt \\ \vdots \\ dc_{2^N-1}/dt \end{pmatrix} = \hat{H}(t) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^N-1} \end{pmatrix} \quad (5.40)$$

で与えられる。ここで

$$\begin{aligned} \hat{H}(t) &= A(t)\Gamma\hat{H}_A(t) + B(t)\hat{H}_B \\ A(t) &= 1 - \frac{t}{\tau} \quad B(t) = \frac{t}{\tau} \\ \hat{H}(0) &= \hat{H}_A \quad \hat{H}(\tau) = \hat{H}_B \end{aligned} \quad (5.41)$$

とする。 $\hat{H}_A(t)$ と $\hat{H}_B(t)$ は

$$\hat{H}_A = \Gamma \begin{pmatrix} 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \end{pmatrix}, \quad \hat{H}_B = \begin{pmatrix} E_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & E_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & E_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & E_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & E_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_7 \end{pmatrix} \quad (5.42)$$

で与えられる。

初期条件を

$$|\psi\rangle_{t=0} = \frac{1}{\sqrt{2^3}} (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T \quad (5.43)$$

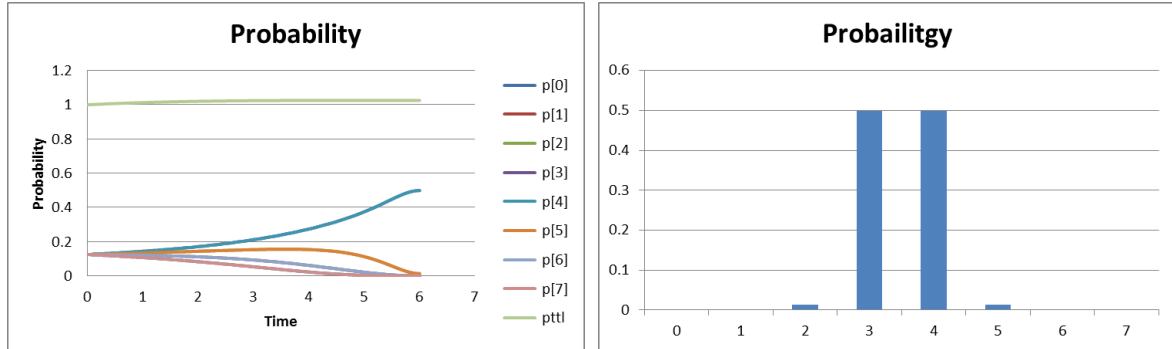
として、式(5.40)で与えられる常微分方程式をオイラー法で解いた。以下のパラメーターを計算に用いた：

$$a = 0.0, \quad \hbar = 0.1, \quad \Gamma = 1.0, \quad dt = 0.00001, \quad \tau = 6.0, \quad L = 1.75 \quad (5.44)$$

計算結果を図 5.4 に示す. 連続変数 x の離散近似は, 式 (5.35) から

$$x_n = -1.75 + n \left(\frac{3.5}{7} \right), \quad n = 0, 1, \dots, 7 \quad (5.45)$$

となる. この場合の目的関数の最小値は, $x = -0.25$ と $x = 0.25$ すなわち基底 $(0,1,1)$ と $(1,0,0)$ で, 同じ値の最小値になる. 図 5.6 に示すように, 基底 $(0,1,1)$ と $(1,0,0)$ で確率が 0.5 になっている.



Probability of base 0, 1, 2, 3, 4, 5, 6 and 7

図 5.4 Solution of $\min_{p=0,1,\dots,7} (x-a)^2$ by quantum annealing ($N_q = 3$; $p = 0, 1, 2, 3, 4, 5, 6, 7$)

(3) 4Q ビット, 任意整数数列の最小値問題

基底のエネルギー E_p ($N_q = 4$; $p = 0, 1, 2, \dots, 15$) は, 関数値ではなくて単なる数列であってよい. 例えば, 表 5.1 および図 5.5 に示されるようなものであって良い.

表 5.1 Setting of Energy E_p ($N_q = 4$; $p = 0, 1, 2, \dots, 15$)

p	E_p	p	E_p
0	2	8	2
1	0.8	9	0.8
2	0	10	1.3
3	1.2	11	1.6
4	1.5	12	1.7
5	0.7	13	1.2
6	0.6	14	0.6
7	1.1	15	0.8

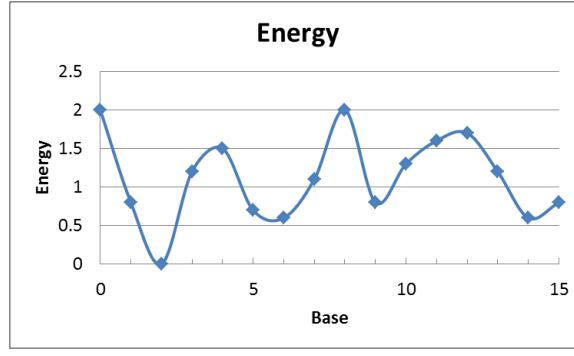


図 5.5 Setting of E_p for quantum annealing ($N=4$; $p=0,1,2,\dots,15$)

ハミルトニアン H_B は, Code 1 を用いて式 (5.42) のような対角行列を作ればよい. H_A については, $N_q=4$ の場合のものを式 (5.42) に倣って作ればよい. 本論では, Code 1 を用いて計算した.

このコードで計算された $N_q=4$ の場合のハミルトニアン H_A を以下に示す:

$H_A =$

$$\begin{array}{cccccccccccccccc}
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\
 -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\
 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0
 \end{array}
 \tag{5.46}$$

このようなハミルトニアン H_A と H_B を用いて, 式 (5.40) と式 (5.41) により時間進化を計算すればよい.

計算パラメーターは

$$\hbar = 0.1, \quad \Gamma = 1, \quad dt = 0.00002, \quad \tau = 6 \tag{5.47}$$

とした. 図 5.6 に計算結果を示す. 確率の計算値の合計が 1 を超えているが, ハミルトニアン $H(t) = (1-t/\tau)\Gamma H_A + (t/\tau)H_B$ が時間 t の関数であるから, 確率は時間の関数であるので合計が 1 を超えても仕方がない. 改めて正規化すればよい. 計算結果を図 5.8 に示す.

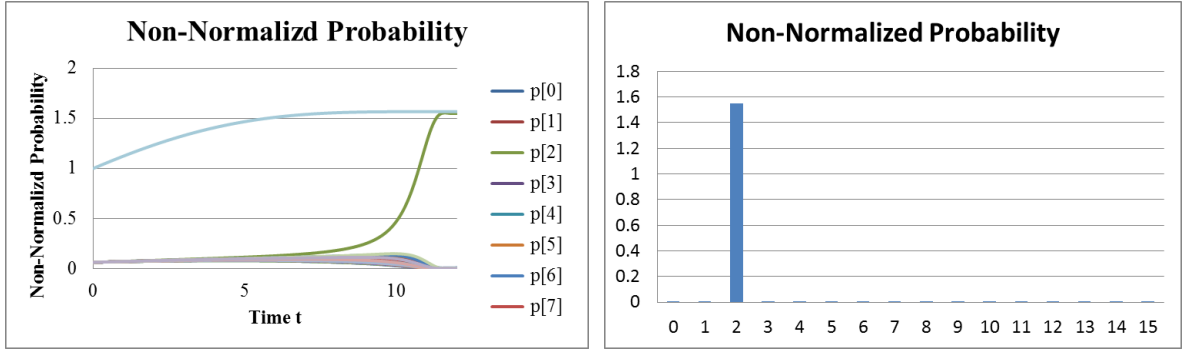


図 5.6 Solution of $\min_{p=0,1,2,\dots,15} f(p)$ by quantum annealing ($N_q = 4$; $p = 0, 1, 2, \dots, 15$)

5.3.2 多変数の場合の量子アニーリング計算

N_q 量子ビット, N_v 連続変数, 任意の関数 $f_x(\vec{x}) = f_x(x_0, x_1, \dots, x_{N_v-1})$ の最小値問題への拡張について考え, 数値例も示す.

N_v 個の連続変数の関数 :

$$f_x(\vec{x}) = f_x(x_0, x_1, \dots, x_{N_v-1}) \quad (5.48)$$

の最小値問題 :

$$f_x(\vec{x}) = f_x(x_0, x_1, \dots, x_{N_v-1}) = \min \quad (5.49)$$

を考える. 簡単のため, 個々の変数 x_i の量子ビット数 N_{q_i} は

$$N_{q_i} = N_q / N_v, \quad i = 0, 1, \dots, N_v - 1 \quad (5.50)$$

とする.

連続変数は離散変数で近似する :

$$\vec{x} = -\vec{X}_L + i\Delta\vec{x} \quad (5.51)$$

整数変数は

$$\Delta x_i = (\Delta\vec{x})_i = 1, \quad i = 0, 1, \dots, N_v - 1 \quad (5.52)$$

と考える. 式 (5.51) における 10 進整数 i は

$$b(0) = 0 \quad \text{and} \quad b(i) = \sum_{j=0}^{i-1} N_{q_i} \quad \text{for } i > 0 \quad (5.53)$$

として, 2 進数に変換する :

$$i = \sum_{n=0}^{N_{q_i}-1} 2^n q_{b(i)+n} = 2^0 q_{b(i)+0} + 2^1 q_{b(i)+1} + \dots + 2^{N_{q_i}-1} q_{b(i)+N_{q_i}-1} \quad (5.54)$$

これを式 (5.51) に代入すると

$$\vec{x} = -\vec{X}_L + \left(2^0 q_{b(i)+0} + 2^1 q_{b(i)+1} + \dots + 2^{N_{q_i}-1} q_{b(i)+N_{q_i}-1} \right) \Delta\vec{x} \quad (5.55)$$

を得る. これを式 (5.49) に代入すると, 関数 f_x は N_q 個の 2 値変数 $q_0, q_1, \dots, q_{N_q-1}$ の関数 f_q に変換される.

すなわち, 式 (5.49) は

$$f_x(\vec{x}) = f_q(q_0, q_1, \dots, q_{N_q-1}) = \min \quad (5.56)$$

に変換される.

2 値変数空間の基底の数 N_b は

$$N_b = 2^{N_q} \quad (5.57)$$

で与えられる. 具体的には 10 進数 p を

$$p = \sum_{i=0}^{N_q} 2^i q_i^p = 2^0 q_0^p + 2^1 q_1^p + \cdots + 2^{N_q-1} q_{N_q-1}^p \quad (5.58)$$

とするとき, p 番目の基底を

$$\begin{pmatrix} q_0^p & q_1^p & \cdots & q_{N_q-1}^p \end{pmatrix} \quad (5.59)$$

とすると, このときの基底エネルギー E_p は

$$E_p = f_q(q_0^p, q_1^p, \cdots, q_{N_q-1}^p) \quad (5.60)$$

となる.

以下に数値計算例を示す.

(1) 4Q ビット, 2 整数変数, 2 次関数の最小値問題

目的関数として 2 変数の場合を考える:

$$f(i, j) = \alpha(i-a)^2 + \beta(i-a)(j-b)j + \gamma(j-b)^2 \quad (5.61)$$

ここで

$$i = \sum_{\rho=0}^1 2^\rho q_\rho = q_0 + 2q_1, \quad j = \sum_{\rho=2}^3 2^{\rho-2} q_\rho = q_2 + 2q_3 \quad (5.62)$$

4Q ビットの場合のエネルギーは

$$\begin{aligned} E_0 &= f_{ij}(0,0) = E_{0000} = f_q(0,0,0,0), & E_1 &= f_{ij}(1,0) = E_{0100} = f_q(0,1,0,0), \\ E_2 &= f_{ij}(2,0) = E_{1000} = f_q(1,0,0,0), & E_3 &= f_{ij}(3,0) = E_{1100} = f_q(1,1,0,0), \\ E_4 &= f_{ij}(0,1) = E_{0001} = f_q(0,0,0,1), & E_5 &= f_{ij}(1,1) = E_{0101} = f_q(0,1,0,1), \\ E_6 &= f_{ij}(2,1) = E_{1001} = f_q(1,0,0,1), & E_7 &= f_{ij}(3,1) = E_{0101} = f_q(1,1,0,1), \\ E_8 &= f_{ij}(0,2) = E_{0010} = f_q(0,0,1,0), & E_9 &= f_{ij}(1,2) = E_{0110} = f_q(0,1,1,0), \\ E_{10} &= f_{ij}(2,2) = E_{1010} = f_q(1,0,1,0), & E_{11} &= f_{ij}(3,2) = E_{1110} = f_q(1,1,1,0), \\ E_{12} &= f_{ij}(0,3) = E_{0011} = f_q(0,0,1,1), & E_{13} &= f_{ij}(1,3) = E_{0111} = f_q(0,1,1,1), \\ E_{14} &= f_{ij}(2,3) = E_{1011} = f_q(1,0,1,1), & E_{15} &= f_{ij}(3,3) = E_{1111} = f_q(1,1,1,1), \end{aligned} \quad (5.63)$$

とすればよい. ハミルトニアン H_B は, 式 (5.63) を計算すれば簡単に作れる. ハミルトニアン H_A は式 (5.46) を使えばよい.

このようなハミルトニアン H_A と H_B を用いて, 式 (5.40) と式 (5.41) により時間進化を計算すればよい. 計算パラメータは

$$\hbar = 0.1, \quad \Gamma = 1, \quad dt = 0.00001, \quad \tau = 6; \cdots \alpha = 1, \quad \beta = 0, \quad \gamma = 1; \quad a = 1, \quad b = 1 \quad (5.64)$$

とした. 図 5.7 に計算結果を示す.

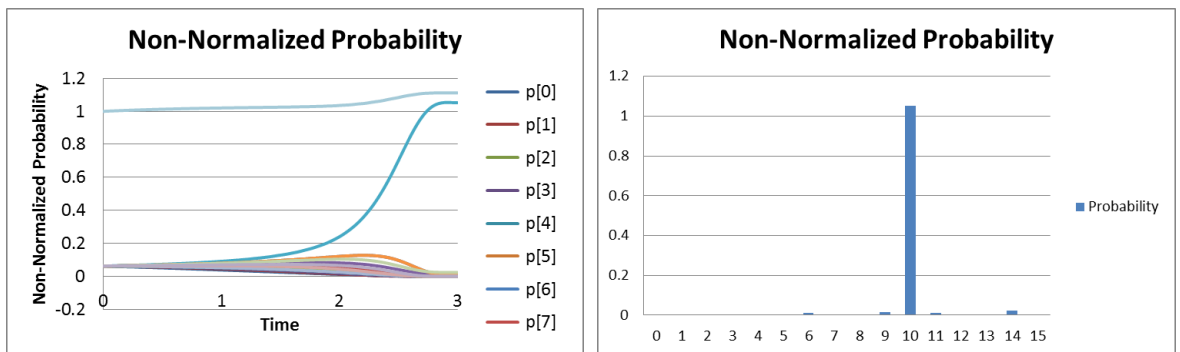


図 5.7 Solution of $\min_{i=0,1,2,3; j=0,1,2,3} f(i, j)$ by quantum annealing ($N_q = 4; i = 0, 1, 2, 3; j = 0, 1, 2, 3$)

5.3.3 量子アニーリング計算の特長

量子アニーリングは基底エネルギーの最小値を、量子力学の原理を利用して求めるものであり、答えは測定時の基底の係数すなわち基底の出現確率に従って与えられる。他の最小値探索とは異なり、確実に大域的最小値を計算できる。

(1) 明確な結論が出る

大域的最小値と局所的最小値の間に明確な差がある場合には問題ないが、極めて近い値の局所的最小値がある場合には、大域的最小値との出現確率の差が縮まるので注意が必要であるが明確に区別できる。図 5.8 に計算結果を示す。計算は § 5.3.1 (3) のケースを少し変更したものについて行った。図 5.6 に結果を示す。D-Wave の専用機では同じ計算を 1000 回繰り返し、大域的最小値の推定値の頻度分布を求めて大域的最小値を推定している。その理由は誤判断を防ぐためであろう。

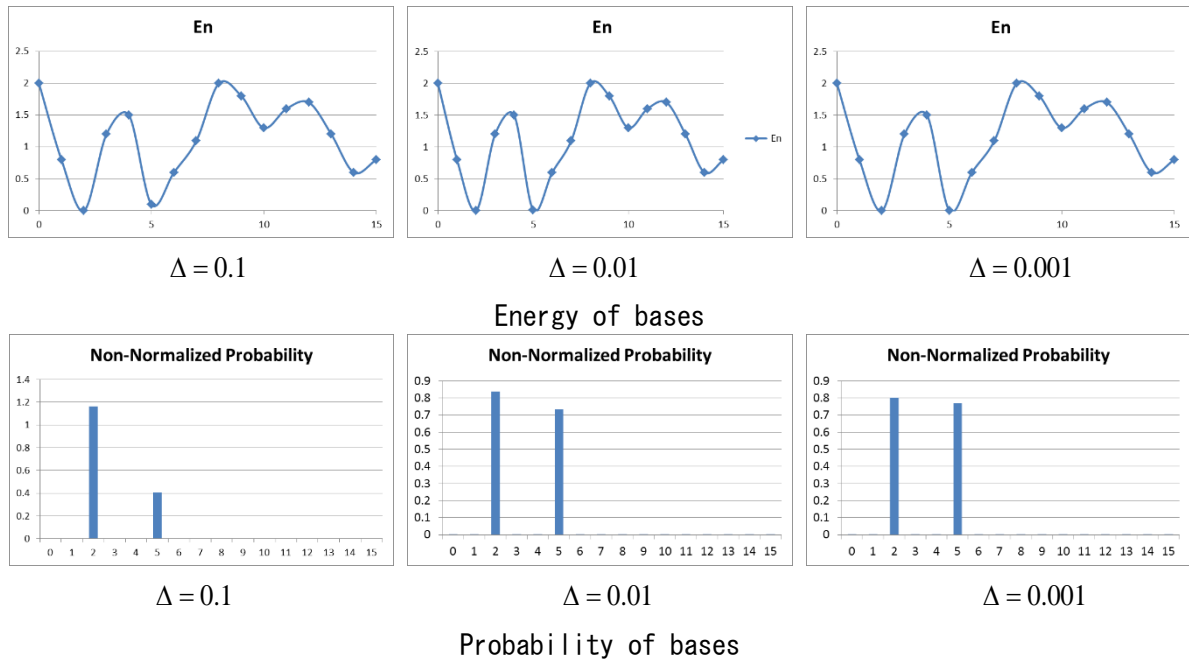


図 5.8 Effects of local minimums on the global minimum

(2) ノイズに強い

D-Wave の専用機の量子アニーリングは、ジョセフソン素子のコヒーレンス時間の観点から考えると、1 回の計算にコヒーレンス時間(数 10 ナノ秒)を優に超える長大な時間(数マイクロ秒)を掛けている[西森・大関]。このことは、量子アニーリングがノイズに強いという性質があることを示唆しているように思われる。そこで、ノイズ項 $(\varepsilon_0 \ \varepsilon_1 \ \dots \ \varepsilon_{2^{N_q}-1})^T$ を付加した時間発展計算を行ってみた：

$$i\hbar \begin{pmatrix} dc_0/dt \\ dc_1/dt \\ \vdots \\ dc_{2^{N_q}-1}/dt \end{pmatrix} = \hat{H}(t) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^{N_q}-1} \end{pmatrix} + \sigma \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_{2^{N_q}-1} \end{pmatrix} \quad (5.65)$$

ここで

$$\begin{aligned}\hat{H}(t) &= A(t)\Gamma\hat{H}_A(t) + B(t)\hat{H}_B \\ A(t) &= 1 - \frac{t}{\tau} \quad B(t) = \frac{t}{\tau} \\ \hat{H}(0) &= \hat{H}_A \quad \hat{H}(\tau) = \hat{H}_B\end{aligned}\tag{5.66}$$

計算は3.1 (3) のケースについて行った。ノイズは $[-1,1]$ 一様分布ノイズで、計算結果を図5.9に示す。ノイズの影響が小さいことが分かる。

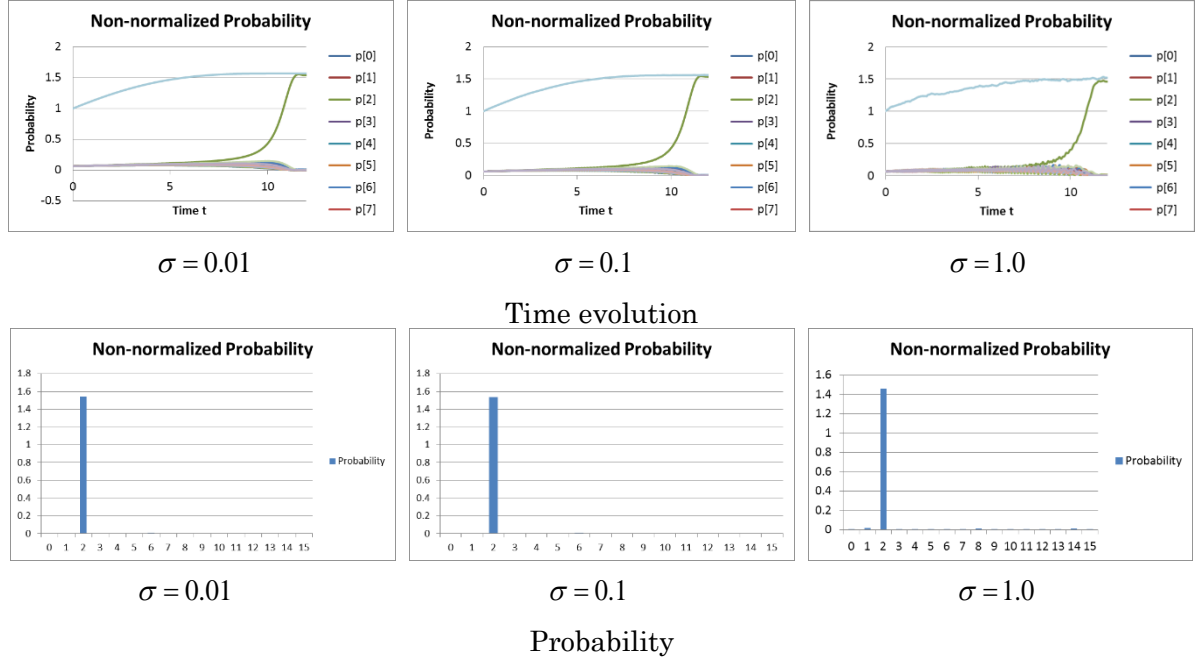


図 5.9 Effects of noise

5.4 拘束条件のある場合の最小値問題

5.4.1 等式拘束条件のある場合

等式拘束条件の場合は簡単である。等式拘束条件をペナルティーとして付け加えるだけで良い。すなわち

$$f(x, y) = \min \quad \text{under } g(x, y) = 0\tag{5.67}$$

は $\eta > 0$ をペナルティーとして

$$F(x, y, \eta) = f(x, y) + \eta g^2(x, y) = \min\tag{5.68}$$

を考えればよい。

例えば,

$$f(x, y) = x + y = \min \quad \text{under } g(x, y) = x^2 + y^2 - 1 = 0\tag{5.69}$$

の場合には

$$F(x, y) = x + y + \eta(x^2 + y^2 - 1)^2 = \min\tag{5.70}$$

となる。 $\eta=1$ とすると

$$\frac{\partial F}{\partial x} = 1 + 4(x^2 + y^2 - 1)x = 0, \quad \frac{\partial F}{\partial y} = 1 + 4(x^2 + y^2 - 1)y = 0\tag{5.71}$$

であるので

$$x = y\tag{5.72}$$

これを拘束条件に代入すると

$$2x^2 = 1 \rightarrow x = \pm \frac{1}{\sqrt{2}} \rightarrow y = \pm \frac{1}{\sqrt{2}} \quad (5.73)$$

を得る。答えとして4個の可能性があるが

$$x = y = -\frac{1}{\sqrt{2}} \quad (5.74)$$

が最小値 $-\sqrt{2}$ を与える。

5.4.2 不等式拘束条件のある場合

不等式拘束条件がある場合、例えば

$$f(x, y) = \min \quad \text{under } g(x, y) \geq 0 \quad (5.75)$$

の場合には、最小点が不等式拘束条件が定義する領域内にある場合には、最小点は

$$\frac{\partial f(x, y)}{\partial x} = \frac{\partial f(x, y)}{\partial y} = 0 \quad (5.76)$$

を満足する。

領域の境界上にある場合には、曲面 $z = f(x, y)$ 曲面と $z = g(x, y)$ の等高線が図 5.10 に示されるような関係にある場合には、最小点は境界：

$$g(x, y) = 0 \quad (5.77)$$

の上にあって

$$\nabla f(x, y) = -\nabla g(x, y) \quad (5.78)$$

を満足する。

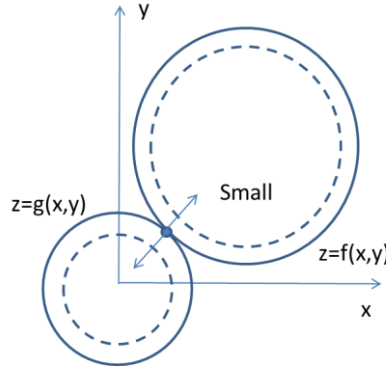


図 5.10 The direction of gradients of surface $z = f(x, y)$ and surface $z = g(x, y)$

従って、最小値問題は

$$f(x, y) = \min \quad \text{under } \nabla f(x, y) + \lambda \nabla g(x, y) = 0 \quad (5.79)$$

に変換される。

故に、 $\eta_1 > 0$ および $\eta_2 > 0$ をペナルティーとして

$$F(x, y) = f(x, y) + \eta_1 g^2(x, y) + \eta_2 \|\nabla f(x, y) + \lambda \nabla g(x, y)\|^2 = \min \quad (5.80)$$

を考えればよい。

例えば、

$$f(x, y) = (x - \sqrt{2})^2 + (y - \sqrt{2})^2 = \min \quad \text{under } g(x, y) = x^2 + y^2 - 1 \geq 0 \quad (5.81)$$

の場合には

$$F(x, y) = (x - \sqrt{2})^2 + (y - \sqrt{2})^2 + \eta_1 (x^2 + y^2 - 1)^2 + \eta_2 \left((2(x - \sqrt{2}) + 2\lambda x)^2 + (2(y - \sqrt{2}) + 2\lambda y)^2 \right) = \min \quad (5.82)$$

となる。ここで

$$\begin{aligned} \partial 4\eta_2 \left((x - \sqrt{2}) + \lambda x \right)^2 / \partial x &= 8\eta_2 \left((x - \sqrt{2}) + \lambda x \right) (1 + \lambda) = 8\eta_2 \left((1 + \lambda)x - \sqrt{2} \right) (1 + \lambda) \\ \partial 4\eta_2 \left((y - \sqrt{2}) + \lambda y \right)^2 / \partial y &= 8\eta_2 \left((y - \sqrt{2}) + \lambda y \right) (1 + \lambda) = 8\eta_2 \left((1 + \lambda)y - \sqrt{2} \right) (1 + \lambda) \end{aligned} \quad (5.83)$$

であるので、 $\eta = 1$ とすると

$$\begin{aligned} \frac{\partial F}{\partial x} &= 2(x - \sqrt{2}) + 4\eta_1 (x^2 + y^2 - 1)x + 8\eta_2 \left((1 + \lambda)x - \sqrt{2} \right) (1 + \lambda) = 0, \\ \frac{\partial F}{\partial y} &= 2(y - \sqrt{2}) + 4\eta_1 (x^2 + y^2 - 1)y + 8\eta_2 \left((1 + \lambda)y - \sqrt{2} \right) (1 + \lambda) = 0 \end{aligned} \quad (5.84)$$

となる。従って

$$\begin{aligned} \frac{\partial F}{\partial x} &= 2(x - \sqrt{2}) - 8\eta_2 \sqrt{2} (1 + \lambda) + 4\eta_1 (x^2 + y^2 - 1)x + 8\eta_2 (1 + \lambda)^2 x = 0, \\ \frac{\partial F}{\partial y} &= 2(y - \sqrt{2}) - 8\eta_2 \sqrt{2} (1 + \lambda) + 4\eta_1 (x^2 + y^2 - 1)y + 8\eta_2 (1 + \lambda)^2 y = 0 \end{aligned} \quad (5.85)$$

と書き換えて、上の式に y を掛け下の式に x を掛けて引くと

$$2(x - \sqrt{2})y - 8\eta_2 \sqrt{2} (1 + \lambda)y = 2(y - \sqrt{2})x - 8\eta_2 \sqrt{2} (1 + \lambda)x \rightarrow x = y \quad (5.86)$$

を得るので、 $g(x, y) = x^2 + y^2 = 1$ から

$$x = y = \frac{1}{\sqrt{2}} \quad (5.87)$$

を得る。これより、最小値は 1 と求まる。

この問題に量子アニーリングを適用するためには、例えば 6 量子ビットの場合には

$$x_i = 0.5 + \Delta x (q_0 + 2q_1), \quad y_j = 0.5 + \Delta y (q_2 + 2q_3), \quad \lambda_k = 0.5 + \Delta \lambda (q_4 + 2q_5) \quad (5.88)$$

とすると、10 進連続変数を 2 進整数で離散値に近似できる。基底のエネルギーは

$$\begin{aligned} E_0 &= F_q(0, 0, 0, 0, 0, 0), \quad E_1 = F_q(0, 0, 0, 0, 1, 0), \\ E_2 &= F_q(0, 0, 0, 0, 1, 1), \quad E_3 = F_q(0, 0, 0, 1, 0, 0), \\ &\vdots \\ E_{62} &= F_q(0, 1, 1, 1, 1, 1), \quad E_{63} = F_q(1, 1, 1, 1, 1, 1) \end{aligned} \quad (5.89)$$

で計算できる。

時間進化に必要なシュレディンガーの方程式は、式 (5.40) と式 (5.41) で与えられる。計算結果を図 5.11 に示す。大域的最小値の近似値が計算されている。

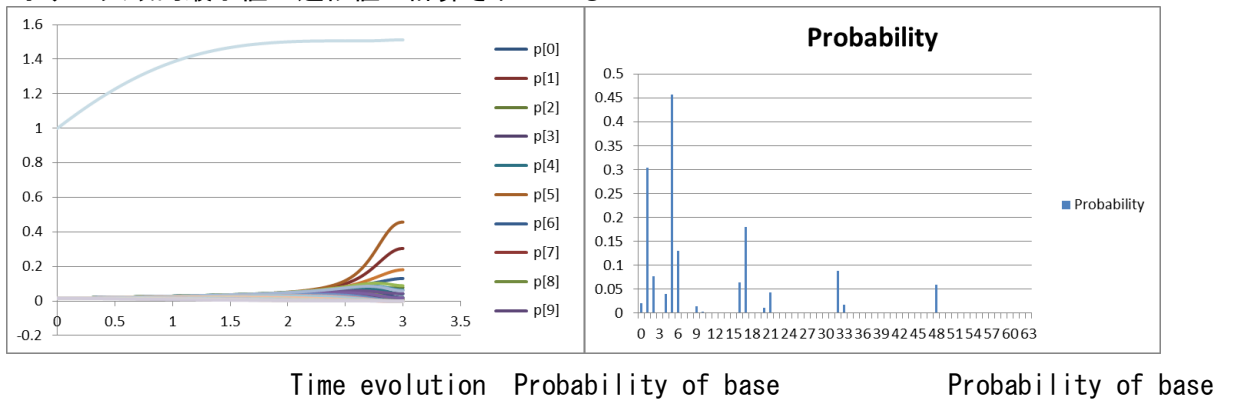


図 5.11 Probability of base

5.5 量子アニーリングによるニューラル・ネットワークの学習

ディープラーニングが開発されて以来、ニューラル・ネットワークの発展には目を見張るものがある。あらゆる分野に甚大な影響を及ぼしつつある。しかし、その学習には大きな課題がある：

- (1) 時間が掛かる。
- (2) 大域的誤差最小解にたどり着くのが簡単でない。
- (3) 過剰学習の恐れがある。

量子アニーリングは、このような課題を一挙に解決してしまう可能性がある。前節までの議論で、量子アニーリングは、あらゆる最小値問題を解決する力を秘めていることが分かった。したがって、この観点から考えれば、量子アニーリングをニューラル・ネットワークに導入できる。簡単のために、階層型ニューラル・ネットワークの誤差逆伝搬学習に限る。

階層型ニューラル・ネットワークのイメージを図 5.12 に示す。入力層を第 0 層、中間層を第 1, 2, ..., $N-1$ 層、出力層を第 N_L 層とする。

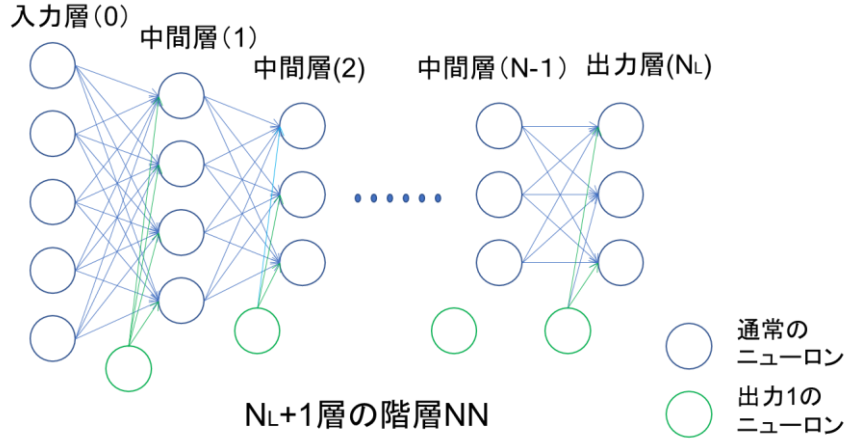


図 5.12 Hierarchical neural network

n 層の j ユニットへの入力は、次式で与えられるものとする：

$$u_j^{(n)} = \sum_{i=1}^{N_O^{(n-1)}} w_{ji}^{(n-1)} v_i^{(n-1)} + \theta_j^{(n-1)}, \quad j=1, 2, \dots, N_O^{(n)}, \quad n=1, 2, \dots, N_L \quad (5.90)$$

ここで、 $w_{ji}^{(n-1)}$ はウェイト、 $v_{si}^{(n-1)}$ は出力、 $\theta_j^{(n-1)}$ はオフセットと呼ばれる。また、 n 層の j ユニットの出力は、次式で与えられるものとする：

$$v_j^{(n)} = h(u_j^{(n)}) = h\left(\sum_{i=1}^{N_O^{(n-1)}} w_{ji}^{(n-1)} v_i^{(n-1)} + \theta_j^{(n-1)}\right), \quad j=1, 2, \dots, N_O^{(n)}, \quad n=1, 2, \dots, N_L \quad (5.91)$$

ここで、 $h(\square)$ は出力関数である。 $v_{si}^{(0)}$ は入力層の入力である。

階層ニューラル・ネットワークは、誤差関数 $E(v, w, \theta)$ が最小になるように学習する：

$$\sum_{s=1}^{N_S} E_s = \sum_{s=1}^{N_S} \left[\frac{1}{2} \sum_{j=1}^{N_O^{(N_L)}} (t_{sj} - v_j^{(N_L)})^2 \right] = \min \quad (5.92)$$

ここで、 N_S は学習サンプル数である。この最小値問題の拘束条件は

$$v_j^{(n)} = h \left(\sum_{i=1}^{N_O^{(n-1)}} w_{ji}^{(n-1)} v_i^{(n-1)} + \theta_j^{(n-1)} \right), \quad j=1,2,\dots,N_O^{(n)}, \quad n=1,\dots,N_L \quad (5.93)$$

で与えられる.

古典計算機で学習する場合には, 誤差逆伝搬と最急降下法を組み合わせた方法で解く. 量子計算機の場合には, 式 (5.93) の拘束条件をペナルティとする最小値問題に変換する. すなわち, 目的関数 F を

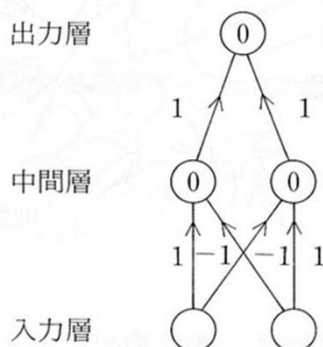
$$F(v^{(N_L)}, \dots, v^{(1)}, \theta^{(N_L-1)}, \dots, \theta^{(0)}, w^{(N_L-1)}, \dots, w^{(0)}) \\ = \sum_{s=1}^{N_S} \left\{ \frac{1}{2} \sum_{j=1}^{N_O^{(N_L)}} (t_j - v_j^{(N_L)})^2 + \sum_{n=1}^{N_L} \sum_{j=1}^{N_O^{(n)}} \eta_j^{(n)} \left[v_j^{(n)} - h \left(\sum_{i=1}^{N_O^{(n-1)}} w_{ji}^{(n-1)} v_i^{(n-1)} + \theta_j^{(n-1)} \right) \right]^2 \right\} = \min \quad (5.94)$$

として量子アニーリングで解く. $\eta_j^{(n)}$, $j=1,2,\dots,N_O^{(n)}$, $n=1,2,\dots,N_L$ はペナルティである.

表 5.2 と図 5.13 に示されるような簡単なニューラル・ネットワークを考えてみる.

表 5.2 Neural network for exclusive OR

出力層	出力	0	1	1	0
	入力	0	1	1	0
中間層	出力	0 0	1 0	0 1	0 0
	入力	0 0	1 -1	-1 1	0 0
入力層	出力	0 0	1 0	0 1	1 1



if $x > 0$ then $h(x) = 1$, else $x = 0$

図 5.13 Neural network for exclusive OR

図 5.13 に示されるようなウェイトが, 式 (5.94) の最小値問題の答えとして得られることを示そう. 式 (5.94) よりこの場合の目的関数 $F(v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, w_{11}^{(0)}, w_{12}^{(0)}, w_{21}^{(0)}, w_{22}^{(0)}, w_{11}^{(1)}, w_{12}^{(1)})$ は

$$F(v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, w_{11}^{(0)}, w_{12}^{(0)}, w_{21}^{(0)}, w_{22}^{(0)}, w_{11}^{(1)}, w_{12}^{(1)})$$

$$= \sum_{s=1}^4 E_s = \sum_{s=1}^{N_s} \left\{ \frac{1}{2} \sum_{j=1}^{N_o^{(2)}} (t_{s,j} - v_j^{(2)})^2 + \sum_{j=1}^1 \eta_j^{(2)} \left[v_j^{(2)} - h \left(\sum_{i=1}^{N_o^{(1)}} w_{ji}^{(1)} v_i^{(1)} + \theta_j^{(1)} \right) \right]^2 \right. \\ \left. + \sum_{j=1}^2 \eta_j^{(1)} \left[v_j^{(1)} - h \left(\sum_{i=1}^{N_o^{(0)}} w_{ji}^{(0)} v_i^{(0)} + \theta_j^{(0)} \right) \right]^2 \right\} = \min \quad (5.95)$$

で与えられる。

量子アニーリングを行うためには、未知変数 $v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, w_{11}^{(0)}, w_{12}^{(0)}, w_{21}^{(0)}, w_{22}^{(0)}, w_{11}^{(1)}, w_{12}^{(1)}$ を 2 進整数変数で近似しなければならない。そのために

$v_1^{(1)}, v_2^{(1)}, v_1^{(2)}$ は

$$v_1^{(1)} = -V_1^{(1)} + \Delta V_1^{(1)} \sum_{\rho=0}^{N_{qv1}^{(1)}-1} 2^\rho q_{b_{v1}^{(1)}+\rho}, \quad v_2^{(1)} = -V_2^{(1)} + \Delta V_2^{(1)} \sum_{\rho=0}^{N_{qv2}^{(1)}-1} 2^\rho q_{b_{v2}^{(1)}+\rho},$$

$$v_1^{(2)} = -V_1^{(2)} + \Delta V_1^{(2)} \sum_{\rho=0}^{N_{qv1}^{(2)}-1} 2^\rho q_{b_{v1}^{(2)}+\rho} \quad (5.96)$$

で近似される。ここで

$$b_{v1}^{(1)} = 0, \quad b_{v2}^{(1)} = b_{v1}^{(1)} + N_{qb1}^{(1)}, \quad b_{v1}^{(2)} = b_{v2}^{(1)} + N_{qv2}^{(1)} \quad (5.97)$$

とする。 $w_{11}^{(0)}, w_{12}^{(0)}, w_{21}^{(0)}, w_{22}^{(0)}, w_{11}^{(1)}, w_{12}^{(1)}$ は

$$w_{11}^{(0)} = -W_{11}^{(0)} + \Delta W_{11}^{(0)} \sum_{\rho=0}^{N_{qw11}^{(0)}-1} 2^\rho q_{b_{w11}^{(0)}+\rho}, \quad w_{12}^{(0)} = -W_{12}^{(0)} + \Delta W_{12}^{(0)} \sum_{\rho=0}^{N_{qw12}^{(0)}-1} 2^\rho q_{b_{w12}^{(0)}+\rho},$$

$$(5.98)$$

$$w_{21}^{(0)} = -W_{21}^{(0)} + \Delta W_{21}^{(0)} \sum_{\rho=0}^{N_{qw21}^{(0)}-1} 2^\rho q_{b_{w21}^{(0)}+\rho}, \quad w_{22}^{(0)} = -W_{22}^{(0)} + \Delta W_{22}^{(0)} \sum_{\rho=0}^{N_{qw22}^{(0)}-1} 2^\rho q_{b_{w22}^{(0)}+\rho}$$

$$w_{11}^{(1)} = -W_{11}^{(1)} + \Delta W_{11}^{(1)} \sum_{\rho=0}^{N_{qw11}^{(1)}-1} 2^\rho q_{b_{w11}^{(1)}+\rho}, \quad w_{12}^{(1)} = -W_{12}^{(1)} + \Delta W_{12}^{(1)} \sum_{\rho=0}^{N_{qw12}^{(1)}-1} 2^\rho q_{b_{w12}^{(1)}+\rho} \quad (5.99)$$

で近似される。ここで

$$b_{w11}^{(0)} = b_{v2}^{(2)} + N_{qv2}^{(2)}, \quad b_{w12}^{(1)} = b_{w11}^{(0)} + N_{qw11}^{(0)}, \quad b_{w21}^{(0)} = b_{w12}^{(1)} + N_{qw12}^{(1)}, \quad b_{w22}^{(0)} = b_{w21}^{(1)} + N_{qw21}^{(1)} \quad (5.100)$$

$$b_{w11}^{(2)} = b_{w22}^{(1)} + N_{qw22}^{(1)}, \quad b_{w12}^{(2)} = b_{w11}^{(2)} + N_{qw11}^{(2)} \quad (5.101)$$

とする。

ハミルトニアン \hat{H}_B を求めるのに必要な各基底のエネルギーは次式で計算される：

$$E_0 = F_q(0,0,0,\dots,0),$$

$$E_1 = F_q(1,0,0,\dots,0),$$

$$E_2 = F_q(0,1,0,\dots,0),$$

$$\vdots$$

$$E_{2^{N_q}-1} = F_q(1,1,1,\dots,1) \quad (5.102)$$

時間進化に必要なシュレディンガーの方程式は、式 (5.40) と式 (5.41) で与えられる。

古典計算機で式 (5.95) で与えられる最小値問題の解を、総当たり法で求めて式 (5.95) の検証を行った。量子アニーラーによる検証は未着手である。

5.6 アニーリングによる連立 1 次方程式の解法

連立 1 次方程式を量子アニーリングで解くアイデアについて述べる。簡単のために 2 元の複素連立 1 次方程式：

$$\begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} \quad (5.103)$$

について考える。ここで、左辺の行列を係数行列、列ベクトルを未知数ベクトルとし、右辺の列ベクトルを定数ベクトルと呼ぶ。

量子アニーリングを使うためには、2 段階の変換：

- (1) 最小値問題に変換
- (2) 10 進変数を 2 進変数に変換

が必要である。

5.6.1 最小値問題に変換

式 (5.56) の両辺に係数行列の共役転置行列を掛ける：

$$\begin{pmatrix} A_{00}^* & A_{10}^* \\ A_{01}^* & A_{11}^* \end{pmatrix} \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} A_{00}^* & A_{10}^* \\ A_{01}^* & A_{11}^* \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} \quad (5.104)$$

* 印は複素共役を表す。係数行列式が 0 でなければ、式 (5.56) と式 (5.57) は等価である。すなわち、式 (5.104) の解は式 (5.103) の解である。

以下の議論を簡単にするために

$$\begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} = \begin{pmatrix} A_{00}^* & A_{10}^* \\ A_{01}^* & A_{11}^* \end{pmatrix} \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix}, \quad \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} = \begin{pmatrix} A_{00}^* & A_{10}^* \\ A_{01}^* & A_{11}^* \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} \quad (5.105)$$

とする。行列 (H_{ij}) は共役転置で変わらないエルミート行列である。式 (5.103) は

$$\begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \quad (5.106)$$

に変換される。

つぎに、式 (5.106) を非負の関数 $F(x_0, x_1)$ を目的関数とする最小値問題：

$$F(x_0, x_1) = \frac{1}{2} \left[\begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} - \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \right]^\dagger \left[\begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} - \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \right] = \min \quad (5.107)$$

に変換する。ここで、上付き添え字 \dagger は共役転置を意味する。式 (5.107) を書き直すと

$$\begin{aligned} F(x_0, x_1) &= \frac{1}{2} \left[\begin{pmatrix} x_0^* & x_1^* \end{pmatrix} \begin{pmatrix} H_{00}^* & H_{10}^* \\ H_{01}^* & H_{11}^* \end{pmatrix} - \begin{pmatrix} g_0^* & g_1^* \end{pmatrix} \right] \left[\begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} - \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \right] \\ &= \frac{1}{2} \left[\begin{pmatrix} x_0^* & x_1^* \end{pmatrix} \begin{pmatrix} H_{00}^* & H_{10}^* \\ H_{01}^* & H_{11}^* \end{pmatrix} \begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \right. \\ &\quad \left. - \begin{pmatrix} g_0^* & g_1^* \end{pmatrix} \begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} - \begin{pmatrix} g_0^* & g_1^* \end{pmatrix} \begin{pmatrix} H_{00}^* & H_{10}^* \\ H_{01}^* & H_{11}^* \end{pmatrix} \begin{pmatrix} x_0^* \\ x_1^* \end{pmatrix} \right. \\ &\quad \left. + \begin{pmatrix} g_0^* & g_1^* \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \right] \\ &= \min \end{aligned} \quad (5.108)$$

行列 (H_{ij}) はエルミートであるので、右辺第 1 項は実数である。

5.6.2 10 進変数を 2 進変数に変換

式の展開を簡潔にするために、つぎのような記号を導入する：

$$\{x\} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \quad [H] = \begin{pmatrix} H_{00} & H_{01} \\ H_{10} & H_{11} \end{pmatrix}, \quad \{g\} = \begin{pmatrix} g_0 \\ g_1 \end{pmatrix} \quad (5.109)$$

この記号を用いると連立 1 次方程式 (5.106) は

$$[H]\{x\} = \{g\} \quad (5.110)$$

最小値問題 (5.108) は

$$F(x_0, x_1) = \frac{1}{2} \left[\{x\}^\dagger [H]^\dagger [H] \{x\} - 2 \operatorname{Re} \left[\{g\}^\dagger [H] \{x\} \right] + \{g\}^\dagger \{g\} \right] = \min \quad (5.111)$$

と書ける。

以下においては、簡単のために複素数ではなくて実数の場合を考える。従って、上付き添え字 \dagger を T と解釈する。10 進変数：

$$x_0 \in \Delta x [-2^{\nu-1}, 2^{2\nu-1} - 1], \quad x_1 \in \Delta x [-2^{\nu-1}, 2^{2\nu-1} - 1] \quad (5.112)$$

を 2 進変数に変換する：

$$\begin{aligned} x_0 &= \Delta x (2^0 q_0 + 2^1 q_1 + \cdots + 2^{\nu-1} q_{\nu-1} - 2^{\nu-1}) \\ x_1 &= \Delta x (2^0 q_\nu + 2^1 q_{\nu+1} + \cdots + 2^{2\nu-1} q_{2\nu-1} - 2^{\nu-1}) \end{aligned} \quad (5.113)$$

ここで、 Δx は分解能で

$$q_\rho = 0, 1 \quad \rho = 0, 1, \dots, \nu-1, \nu, \dots, 2\nu-1 \quad (5.114)$$

は 2 値変数である。式 (5.113) のような変換では、10 進変数が連続ではなくて分解能 Δx の離散 10 進変数を 2 進整数に変換していることになる。ここで次のような記号を導入する：

$$\begin{aligned} \{q\} &= (q_0 \quad q_1 \quad \cdots \quad q_{\nu-1} \quad q_\nu \quad q_{\nu+1} \quad \cdots \quad q_{2\nu-1})^T, \\ [B] &= \begin{pmatrix} 2^0 & 2^1 & \cdots & 2^{\nu-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 2^0 & 2^1 & \cdots & 2^{\nu-1} \end{pmatrix}, \\ \{\Lambda\} &= (2^{\nu-1} \quad 2^{\nu-1})^T \end{aligned} \quad (5.115)$$

式 (5.113) はつぎのように書けることになる：

$$\{x\} = \Delta x ([B]\{q\} - \{\Lambda\}) \quad (5.116)$$

式 (5.115) と式 (5.116) の代わりに

$$\begin{aligned} [Q] &= \begin{pmatrix} q_{00} & q_{01} & \cdots & q_{02} \\ q_{10} & q_{11} & \cdots & q_{12} \end{pmatrix}, \\ \{b\} &= (2^0 \quad 2^1 \quad \cdots \quad 2^{\nu-1})^T, \\ \{\Lambda\} &= (2^{\nu-1} \quad 2^{\nu-1})^T \end{aligned} \quad (5.117)$$

として

$$\{x\} = \Delta x ([Q]\{b\} - \{\Lambda\}) \quad (5.118)$$

のように書くことも可能であるが、以下においては式 (5.115) と式 (5.116) を用いる。

式 (5.116) を式 (5.111) に代入すると、目的関数 $F(x_0, x_1) = f(q_0, q_1, \dots, q_{2\nu-1})$ は

$$\begin{aligned}
f(q_0, q_1, \dots, q_{2\nu-1}) &= \frac{1}{2} \begin{bmatrix} \Delta x^2 ([B]\{q\} - \{\Lambda\})^\dagger [H]^\dagger [H] ([B]\{q\} - \{\Lambda\}) \\ -2\Delta x \operatorname{Re}[\{g\}^\dagger [H] ([B]\{q\} - \{\Lambda\})] + \{g\}^\dagger \{g\} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \Delta x^2 \{q\}^\dagger [B]^\dagger [H]^\dagger [H] [B] \{q\} \\ -\Delta x^2 \{\Lambda\}^\dagger [H]^\dagger [H] [B] \{q\} - \Delta x^2 \{q\}^\dagger [B]^\dagger \{\Lambda\} - 2\Delta x \operatorname{Re}[\{g\}^\dagger [H] [B] \{q\}] \\ +\Delta x^2 \{\Lambda\}^\dagger \{\Lambda\} + 2\Delta x \operatorname{Re}[\{g\}^\dagger [H] \{\Lambda\}] + \{g\}^\dagger \{g\} \end{bmatrix} \\
&= \min
\end{aligned} \tag{5.119}$$

となる。これは 2 進変数 $q_\rho = 0, 1$ $\rho = 0, 1, \dots, \nu-1, \nu, \dots, 2(\nu-1)$ の 2 次関数であるので、量子アニーリングが可能である。

5.7 本章のまとめ

本研究では、量子ビットによって超並列計算を可能にする量子アニーリングによって、任意の最小値問題の大域的最小値を求める基本的な数理を検討するとともに、数値例で検証した。

任意の最小値問題の大域的最小値を求めるためには、以下のような課題に取り組まねばならない：

- (1) 10 進整数変数の 2 進整数変数への変換
- (2) 10 進連続変数の 2 進整数変数による近似
- (3) 2 次関数でない一般の関数への拡張

さらに、そのアイデアとアルゴリズムの有効性を実証するために、古典計算機で簡単なシミュレーションを実施した。次のステップにおいて、量子計算機による計算を実施したい。

また、本論では、量子情報処理技術に馴染の薄い読者のために、先ず昨今高い関心を招いているアニーリングと呼ばれる最小値問題の解法の解説を独自のアイデアを盛り込んで行っている。さらに、量子アニーリングによるニューラル・ネットワークの学習と連立 1 次方程式の解法についても、基礎的な数理の検討結果を述べた。

第 5 章の参考文献

- [5-1] T. Kadowaki, H. Nishimori, Quantum Annealing in the Transverse Ising model, Physical Review E, Vol. 58, Iss. 5, 5355 (1998.11).
- [5-2] 西森秀稔, 量子アニーリングの解説, (最終更新 2022/3/23).
- <http://q-annealing.org/QA/q-annealing.html>
- [5-3] 西森秀稔, 大関真之, 量子アニーリングの基礎, 共立出版 (2018).
- [5-4] 田中 宗, 田村 亮, Bikas K. Chakrabarti, 量子アニーリングの物理, 森北出版 (2023).
- [5-5] Arnab Das, Bikas K. Chakrabarti [Eds], Quantum Annealing and Related Optimization Methods, Springer (2010).
- [5-6] FIXSTARS Amplify, 量子アニーリングの原理.
- <https://amplify.fixstars.com/ja/techresources/annealing-method/ising-model/principle/>
- [5-7] NTT データ量子コンピューティングガイドライン NTT アニーリングイジングマシン編 NTT DaTa (2021/01).
- https://www.nttdata.com/jp/ja/-/media/nttdatajapan/files/news/services_info/2021/012800/012800-01.pdf

第 5 章の付録 A C 言語で書かれた量子アニーリングの計算コード

以下の手順で計算できる。

- (1) gnu の C コンパイラをインストールする。
- (2) 計算のためのフォルダーを作る。例えば **desktop** の下に **quantumAnnealing** を作る。
- (3) **quantumAnnealing** に下記の source code のファイル **qubo1dimParab1Y.c** と入力ファイル **qubo1dimParab1Y_inp.dat** を置く。
- (4) コマンドウィンドウを開く。
- (5) コマンドウィンドウのディレクトリを `>%users%\1%desktop%\quantumAnnealing` にする。
- (6) ここで `>gcc qubo1dimParab1Y.c -o qubo1dimParab1Y.exe` をタイプして CR すると、実行ファイル **qubo1dimParab1Y.exe** とオブジェクトファイル **qubo1dimParab1Y.obj** がフォルダー **quantumAnnealing** にできる。オブジェクトファイルは不要なので切り捨ててよい。
- (7) フォルダー **quantumAnnealing** で実行ファイル **qubo1dimParab1Y.exe** をクリックすると出力ファイル **qubo1dimParab1Y_out.csv** が出力される。
- (8) 出力ファイル **qubo1dimParab1Y_out.csv** をクリックするとエクセルが開き計算結果を図にできる。

(1) 3Q ビット, 1 整数変数, 2 次関数の最小値問題

ソースファイル : qubo1dimParab1Y.c

```
////////////////////////////////////
//
//      ProgramMulti   ising4.c
//
//
//
//
//      File           ising4.c 2023.02.08-02.11 //
//      File    qubo1dimParab.c 2023.03.24-03.24 //
//      File    qubo1dimParab1.c 2023.03.25-03.25 //
//      File    qubo1dimParab1Y.c 2023.03.25-04.30 //
//
//
////////////////////////////////////

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define Pm 2.0e-08
#define PI 3.14159265 // Pi

// ----- //

void main();
void pushKey();
```

```

double drand();           // uniform random number in (-1,1)
double drand01();         // uniform random number in (0,1)
double normal();          // normal distri

double E(int,int, int, double); // object function

void GenOfHA(int);        // generaton of HA matrix
double Rvs(double);       // if 0, -1; if -1, then 0

// ----- //

double a;                //

int Nq;                  // number of qubit
int Nb;                  // number of base

double hb;               //

double t[300001];        // time
double tau;              //
double dt;               //

double G;                // Gamma: coefft. for stirring Hamiltonian

double HA[8][8];         // Stirring Hamiltonian
double HB[8][8];         // Problem specific Hamiltonian
double H[8][8];          // Hamiltonian

double cc[8][300001];    // coeffts. of base
double cs[8][300001];    //

double p[8][300001];     // probability
double pttl[300001];     //

long NT;                 //

FILE *fp_inp;            // pointer of input file
FILE *fp_out;            // pointer of output file

char InputDataFile[80];  // input file name
char OutputDataFile[80]; // output file name

char buf[5000];

// ----- //

void main()
{
    long ii;
    int i, j, k;
    int m, n;
    double tmp;

```

```

// Input file
sprintf(InputDataFile, "qubo1dimParab2Y_inp.dat");

if ((fp_inp = fopen(InputDataFile, "r")) == NULL) {
    printf("Failed in Reading Input Data File! ... %s\n", InputDataFile);
    fprintf(fp_out, "Failed in Reading Input Data File! ... %s\n", InputDataFile);
    exit(1);
}

// Output file
sprintf(OutputDataFile, "qubo1dimParab2Y_out.csv");

if ((fp_out = fopen(OutputDataFile, "w")) == NULL) {
    printf("Failed in Reading Output Data File! ... %s\n", OutputDataFile);
    fprintf(fp_out, "Failed in Reading Output Data File! ... %s\n",
        OutputDataFile);
    exit(1);
}

// input data from file //
fscanf(fp_inp, "%s %d", buf, &Nq);
fscanf(fp_inp, "%s %lf", buf, &a);
fscanf(fp_inp, "%s %lf", buf, &hb);
fscanf(fp_inp, "%s %lf", buf, &G);

fscanf(fp_inp, "%s %lf", buf, &dt);
fscanf(fp_inp, "%s %ld", buf, &NT);

Nb = (int)powf(2.0, Nq+0.0);

printf("Nq      = %d\n", Nq);
printf("Nb      = %d\n", Nb);
printf("a       = %12.6f\n", a);
printf("hb      = %12.6f\n", hb);
printf("G       = %12.6f\n", G);

printf("dt      = %12.6g\n", dt);
printf("NT      = %ld\n", NT);
printf("\n");

fprintf(fp_out, "Nq =, %d\n", Nq);
fprintf(fp_out, "Nb =, %d\n", Nb);
fprintf(fp_out, "a =, %12.6f\n", a);
fprintf(fp_out, "hb =, %12.6f\n", hb);
fprintf(fp_out, "G =, %12.6f\n", G);

fprintf(fp_out, "dt =, %12.6g\n", dt);
fprintf(fp_out, "NT =, %ld\n", NT);
fprintf(fp_out, "\n");

```



```

///// seach for global minimum

tau = (NT+0.0)*dt;

for (m = 0; m < Nb; m++)
    cc[m][0] = 1.0/sqrt(Nb+0.0);

for (m = 0; m < Nb; m++)
    cs[m][0] = 0.0/sqrt(Nb+0.0);

for (i = 0; i <= NT; i++)
    t[i] = (i+0.0)*dt;

for (n = 0; n < Nb; n++)
    p[n][0] = cc[n][0]*cc[n][0]+cs[n][0]*cs[n][0];
pttl[0] = 0.0;
for (n = 0; n < Nb; n++)
    pttl[0] += p[n][0];

GenOfHA(Nq);

fprintf(fp_out, "HA[i][j]¥n");
fprintf(fp_out, "''", "");
for (j = 0; j < Nb; j++)
    fprintf(fp_out, "j = %d, ", j);
fprintf(fp_out, "¥n");
for (i = 0; i < Nb; i++) {
    fprintf(fp_out, "i = %d, ", i);
    for (j = 0; j < Nb; j++)
        fprintf(fp_out, "%12.6f, ", HA[i][j]);
    fprintf(fp_out, "¥n");
}
fprintf(fp_out, "¥n");

for (i = 0; i < Nb; i++)
    for (j = 0; j < Nb; j++)
        HB[i][j] = 0.0;

HB[0][0] = E(0, 0, 0, a);
HB[1][1] = E(1, 0, 0, a);
HB[2][2] = E(0, 1, 0, a);
HB[3][3] = E(1, 1, 0, a);
HB[4][4] = E(0, 0, 1, a);
HB[5][5] = E(1, 0, 1, a);
HB[6][6] = E(0, 1, 1, a);
HB[7][7] = E(1, 1, 1, a);

fprintf(fp_out, "HB[i][j]¥n");

```

```

fprintf(fp_out, "''', ");
for (j = 0; j < Nb; j++)
    fprintf(fp_out, "j = %d, ", j);
fprintf(fp_out, "¥n");
for (i = 0; i < Nb; i++) {
    fprintf(fp_out, "i = %d, ", i);
    for (j = 0; j < Nb; j++)
        fprintf(fp_out, "%12.6f, ", HB[i][j]);
    fprintf(fp_out, "¥n");
}
fprintf(fp_out, "¥n");

for (ii = 1; ii <= NT; ii++) {

    for (i = 0; i < Nb; i++)
        for (j = 0; j < Nb; j++)
            H[i][j] = G*(1.0-t[ii]/tau)*HA[i][j]+t[ii]/tau*HB[i][j];

    for (m = 0; m < Nb; m++) {
        cc[m][ii] = cc[m][ii-1];
        for (n = 0; n < Nb; n++)
            cc[m][ii] += 1.0/hb*H[m][n]*cs[n][ii-1]*dt;
    }

    for (m = 0; m < Nb; m++) {
        cs[m][ii] = cs[m][ii-1];
        for (n = 0; n < Nb; n++)
            cs[m][ii] -= 1.0/hb*H[m][n]*cc[n][ii-1]*dt;
    }

    for (n = 0; n < Nb; n++)
        p[n][ii] = cc[n][ii]*cc[n][ii]+cs[n][ii]*cs[n][ii];
    pttl[ii] = 0.0;
    for (n = 0; n < Nb; n++)
        pttl[ii] += p[n][ii];
}
fprintf(fp_out, "¥n");

fprintf(fp_out, "ii, t, ");
for (n = 0; n < Nb; n++)
    fprintf(fp_out, "cc[%d], cs[%d], ", n, n);
for (n = 0; n < Nb; n++)
    fprintf(fp_out, "p[%d], ", n);
fprintf(fp_out, "pttl¥n");

for (ii = 0; ii <= NT; ii++)
    if (ii%1000 == 0) {
        fprintf(fp_out, "%ld, %12.6f, ", ii, t[ii]);
    }

```

```

        for (n = 0; n < Nb; n++)
            fprintf(fp_out, "%12.6f, %12.6f, ", cc[n][ii], cs[n][ii]);
        for (n = 0; n < Nb; n++)
            fprintf(fp_out, "%12.6f, ", p[n][ii]);
        fprintf(fp_out, "%12.6f¥n", pttl[ii]);
    }

    pushKey();

    fclose(fp_out);
}

// ----- //

void pushKey()
{
    printf("¥n      Push Return Key! ");
    getchar();
    getchar();
}

// ----- //

double drand()
{
    return 2.0*((double)rand())/((double)RAND_MAX)-0.5;
}

// ----- //

double drand01()
{
    return (drand()+1.0)/2.0;
}

// ----- //

double normal(double x, double myu, double sgm)
{
    return 1.0/sqrt(2.0*PI*sgm*sgm)*exp(-(x-myu)*(x-myu)/(2.0*sgm*sgm));
}

// ----- //

double E(int q0, int q1, int q2, double a) // object function
{
    int p;

    p = q0+2*q1+q2*4;

```

```

    return (p-a)*(p-a);
}

// ----- //

void GenOfHA(int Nq) {

    int i, j;

    // upper left
    HA[0][0] = 0.0;
    for (j = 1; j < Nb/2; j++)
        HA[0][j] = Rvs(HA[0][j-1]);

    for (i = 1; i < Nb/2; i++)
        for (j = 0; j < Nb/2; j++)
            HA[i][j] = Rvs(HA[i-1][j]);

    // lower left
    for (i = 0; i < Nb/2; i++)
        for (j = 0; j < Nb/2; j++)
            HA[Nb/2+i][j] = Rvs(HA[i][j]);

    // upper right
    for (i = 0; i < Nb/2; i++)
        for (j = 0; j < Nb/2; j++)
            HA[i][Nb/2+j] = HA[Nb/2+i][j];

    // lower right
    for (i = 0; i < Nb/2; i++)
        for (j = 0; j < Nb/2; j++)
            HA[Nb/2+i][Nb/2+j] = HA[i][j];
}

// ----- //

double Rvs(double x) {

    if ( x == 0.0)
        return -1.0;
    else
        return 0.0;
}

// ----- //

```

入力ファイル (qubo1dimParab2Y_inp.dat)

```

Nq          3

a           1.0
hb          0.1

```

G 1.0
dt 0.00001
NT 300000

出力ファイル (qubo1dimParab2Y_out.csv)

(2) 3Q ビット, 1 連続変数, 2 次関数の最小値問題

Source code: qubo1dimSquareFlt0.c by H. Isshiki

```
/////////////////////////////////////////////////////////////////
//                                                                    //
//    ProgramMulti   ising4.c                                         //
//                                                                    //
//                                                                    //
//                                                                    //
//          File          ising4.c 2023.02.08-02.11 //
//          File   qubo1dimParab.c 2023.03.24-03.24 //
//          File   qubo1dimParab1.c 2023.03.25-03.25 //
//          File   qubo1dimRoot1.c 2023.03.26-04.02 //
//          File   qubo1dimRoot0.c 2023.04.03-04.03 //
//          File   qubo1dimSquare0.c 2023.04.06-04.06 //
//          File qubo1dimSquareFlt0.c 2023.06.06-04.06 //
//                                                                    //
/////////////////////////////////////////////////////////////////

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define Pm 2.0e-08
#define PI 3.14159265 // Pi

// ----- //

void main();
void pushKey();

double drand(); // (-1,1) の一様乱数を与える関数
double drand01(); // (-1,1) の一様乱数を与える関数
double normal(); // normal distri

double E(double, double); // object function

// ----- //

int Nq; // number of qubits
int Nb; // number of base
```

```

double a;                                //

double h1;                                //
double h2;                                //

double hb;                                //

double t[4000001];                        //
double tau;                               //
double dt;                                // t[it] = (it+0.0)*dt
long it;

double L;                                 // left point
double dx;                                // x[i] = L+(i+0.0)*dt
double x[101];                             // x[i] = L+(i+0.0)*dt

double G;                                 // Gamma: coefft. for stirring Hamiltonian

double HA[8][8];                          // Pauli matrix: Stirring Hamiltonian
double HB[8][8];                          // Problem specific Hamiltonian
double H[8][8];                           // Hamiltonian

double cc[8][4000001];                    //
double cs[8][4000001];                    //

double p[8][4000001];                     //
double ptl[2000001];                      //

long NT;                                  //

FILE *fp_inp;                             // pointer of input file
FILE *fp_out;                             // pointer of output file

char InputDataFile[80];                   // input file name
char OutputDataFile[80];                  // output file name

char buf[5000];

// ----- //

void main()
{
    long i, j, k;
    int m, n;
    double tmp;

    // Input file
    sprintf(InputDataFile, "qubo1dimSquareFlt0_inp.dat");

    if ((fp_inp = fopen(InputDataFile, "r")) == NULL) {
        printf("Failed in Reading Input Data File! ... %s\n", InputDataFile);
    }
}

```

```

    fprintf(fp_out, "Failed in Reading Input Data File! ... %s\n", InputDataFile);
    exit(1);
}

// Output file
sprintf(OutputDataFile, "qubo1dimSquareFlt0_out.csv");

if ((fp_out = fopen(OutputDataFile, "w")) == NULL) {
    printf("Failed in Reading Output Data File! ... %s\n", OutputDataFile);
    fprintf(fp_out, "Failed in Reading Output Data File! ... %s\n",
            OutputDataFile);
    exit(1);
}

// input data from file //
fscanf(fp_inp, "%s %d", buf, &Nq);

fscanf(fp_inp, "%s %lf", buf, &a);

fscanf(fp_inp, "%s %lf", buf, &hb);
fscanf(fp_inp, "%s %lf", buf, &G);

fscanf(fp_inp, "%s %lf", buf, &dt);
fscanf(fp_inp, "%s %ld", buf, &NT);

fscanf(fp_inp, "%s %lf", buf, &L);

Nb = (int)powf(2.0, Nq+0.0);
dx = 2.0*L/(Nb-1.0);
for (i = 0; i < Nb; i++)
    x[i] = -L+(i+0.0)*dx;

printf("Nq      = %d\n", Nq);
printf("Nb      = %d\n", Nb);

printf("a        = %12.6f\n", a);

printf("hb      = %12.6f\n", hb);
printf("G        = %12.6f\n", G);

printf("dt      = %12.6f\n", dt);
printf("NT      = %ld\n", NT);

printf("L        = %12.6f\n", L);
printf("dx      = %12.6f\n", dx);
printf("\n");

fprintf(fp_out, "Nq =, %d\n", Nq);
fprintf(fp_out, "Nb =, %d\n", Nb);

```

```

fprintf(fp_out, "a =, %12.6f¥n", a);

fprintf(fp_out, "hb =, %12.6f¥n", hb);
fprintf(fp_out, "G =, %12.6f¥n", G);

fprintf(fp_out, "dt =, %12.6f¥n", dt);
fprintf(fp_out, "NT =, %ld¥n", NT);

fprintf(fp_out, "L =, %12.6f¥n", L);
fprintf(fp_out, "dx =, %12.6f¥n", dx);
fprintf(fp_out, "¥n");

///// seach for global minimum

tau = (NT+0.0)*dt;

for (m = 0; m < Nb; m++)
//      cc[m][0] = 1.0/sqrt(8.0);
      cc[m][0] = 1.0/sqrt(Nb+0.0);

for (m = 0; m < Nb; m++)
//      cs[m][0] = 0.0/sqrt(8.0);
      cs[m][0] = 0.0/sqrt(Nb+0.0);

for (i = 0; i <= NT; i++)
      t[i] = (i+0.0)*dt;

for (n = 0; n < Nb; n++)
      p[n][0] = cc[n][0]*cc[n][0]+cs[n][0]*cs[n][0];
pttl[0] = 0.0;
for (n = 0; n < Nb; n++)
      pttl[0] += p[n][0];

if (Nq == 2) {
      HA[0][0] = 0.0;
      HA[0][1] = -1.0;
      HA[0][2] = -1.0;
      HA[0][3] = 0.0;

      HA[1][0] = -1.0;
      HA[1][1] = 0.0;
      HA[1][2] = 0.0;
      HA[1][3] = -1.0;

      HA[2][0] = -1.0;
      HA[2][1] = 0.0;
      HA[2][2] = 0.0;
      HA[2][3] = -1.0;

```



```

    HA[3][0] = 0.0;
    HA[3][1] = -1.0;
    HA[3][2] = -1.0;
    HA[3][3] = 0.0;
}
else if (Nq == 3) {
    HA[0][0] = 0.0;
    HA[0][1] = -1.0;
    HA[0][2] = -1.0;
    HA[0][3] = 0.0;
    HA[0][4] = 0.0;
    HA[0][5] = -1.0;
    HA[0][6] = -1.0;
    HA[0][7] = 0.0;

    HA[1][0] = -1.0;
    HA[1][1] = 0.0;
    HA[1][2] = 0.0;
    HA[1][3] = -1.0;
    HA[1][4] = -1.0;
    HA[1][5] = 0.0;
    HA[1][6] = 0.0;
    HA[1][7] = -1.0;

    HA[2][0] = -1.0;
    HA[2][1] = 0.0;
    HA[2][2] = 0.0;
    HA[2][3] = -1.0;
    HA[2][4] = -1.0;
    HA[2][5] = 0.0;
    HA[2][6] = 0.0;
    HA[2][7] = -1.0;

    HA[3][0] = 0.0;
    HA[3][1] = -1.0;
    HA[3][2] = -1.0;
    HA[3][3] = 0.0;
    HA[3][4] = 0.0;
    HA[3][5] = -1.0;
    HA[3][6] = -1.0;
    HA[3][7] = 0.0;

    HA[4][0] = 0.0;
    HA[4][1] = -1.0;
    HA[4][2] = -1.0;
    HA[4][3] = 0.0;
    HA[4][4] = 0.0;
    HA[4][5] = -1.0;
    HA[4][6] = -1.0;
    HA[4][7] = 0.0;

    HA[5][0] = -1.0;

```

```

    HA[5][1] = 0.0;
    HA[5][2] = 0.0;
    HA[5][3] = -1.0;
    HA[5][4] = -1.0;
    HA[5][5] = 0.0;
    HA[5][6] = 0.0;
    HA[5][7] = -1.0;

    HA[6][0] = -1.0;
    HA[6][1] = 0.0;
    HA[6][2] = 0.0;
    HA[6][3] = -1.0;
    HA[6][4] = -1.0;
    HA[6][5] = 0.0;
    HA[6][6] = 0.0;
    HA[6][7] = -1.0;

    HA[7][0] = 0.0;
    HA[7][1] = -1.0;
    HA[7][2] = -1.0;
    HA[7][3] = 0.0;
    HA[7][4] = 0.0;
    HA[7][5] = -1.0;
    HA[7][6] = -1.0;
    HA[7][7] = 0.0;
}
else {
    printf("WRUNG; Nq should be less than or equal to 3!¥n");
    exit(1);
}

for (i = 0; i < Nb; i++)
    for (j = 0; j < Nb; j++)
        if (i == j)
            HB[i][j] = E(x[i], a);
        else
            HB[i][j] = 0.0;

for (it = 1; it <= NT; it++) {

    for (i = 0; i < Nb; i++)
        for (j = 0; j < Nb; j++)
            H[i][j] = G*(1.0-t[it]/tau)*HA[i][j]+t[it]/tau*HB[i][j];

    for (m = 0; m < Nb; m++) {
        cc[m][it] = cc[m][it-1];
        for (n = 0; n < Nb; n++)
            cc[m][it] += 1.0/hb*H[m][n]*cs[n][it-1]*dt;
    }
}

```

```

        for (m = 0; m < Nb; m++) {
            cs[m][it] = cs[m][it-1];
            for (n = 0; n < Nb; n++)
                cs[m][it] -= 1.0/hb*H[m][n]*cc[n][it-1]*dt;
        }

        for (n = 0; n < Nb; n++)
            p[n][it] = cc[n][it]*cc[n][it]+cs[n][it]*cs[n][it];
        pttl[it] = 0.0;
        for (n = 0; n < Nb; n++)
            pttl[it] += p[n][it];
    }
    fprintf(fp_out, "¥n");

    fprintf(fp_out, "it, t, ");
    for (n = 0; n < Nb; n++)
        fprintf(fp_out, "cc[%d], cs[%d], ", n, n);
    for (n = 0; n < Nb; n++)
        fprintf(fp_out, "p[%d], ", n);
    fprintf(fp_out, "pttl¥n");

    for (it = 0; it <= NT; it++)
        if (it%1000 == 0) {
            fprintf(fp_out, "%ld, %12.6f, ", it, t[it]);
            for (n = 0; n < Nb; n++)
                fprintf(fp_out, "%12.6f, %12.6f, ", cc[n][it], cs[n][it]);
            for (n = 0; n < Nb; n++)
                fprintf(fp_out, "%12.6f, ", p[n][it]);
            fprintf(fp_out, "%12.6f¥n", pttl[it]);
        }

    pushKey();

    fclose(fp_out);
}

// ----- //

void pushKey()
{
    printf("¥n      Push Return Key! ");
    getchar();
    getchar();
}

// ----- //

double drand()

```

```

{
    return 2.0*((double)rand())/((double)RAND_MAX)-0.5);
}

// ----- //

double drand01()
{
    return (drand()+1.0)/2.0;
}

// ----- //

double normal(double x, double myu, double sgm)
{
    return 1.0/sqrt(2.0*PI*sgm*sgm)*exp(-(x-myu)*(x-myu)/(2.0*sgm*sgm));
}

// ----- //

double E(double x, double a)                // object function
{
    return (x-a)*(x-a);
}

// ----- //

```

入力ファイル (qubo1dimSquareFlt0_inp.dat)

```

Nq          3

a           0.0
hb          0.1
G           1.25

dt          0.00001
NT          600000

L           1.75

```

出力ファイル (qubo1dimSquareFlt0_out.csv)

(3) 4Q ビット, 任意整数数列の最小値問題

source code ... qubo1dimXHAtestGvnX.c by H. Isshiki

```

//////////////////////////////////////////////////////////////////
//                                                                //
//    ProgramMulti  qubo1dimXHAtestGvnX.c                        //
//                                                                //
//                                                                //

```

```
//
// File ising4.c 2023.02.08-02.11 //
// File qubo1dimParab.c 2023.03.24-03.24 //
// File qubo1dimParab1.c 2023.03.25-03.25 //
// File qubo1dimRoot1.c 2023.03.26-04.02 //
// File qubo1dimRoot0.c 2023.04.03-04.03 //
// File qubo1dimSquare0.c 2023.04.06-04.06 //
// File qubo1dimSquareCvn0.c 2023.04.06-04.06 //
// File qubo1dimXHAtestGvn.c 2023.04.07-04.12 //
// File qubo1dimXHAtestGvnX.c 2023.04.13-04.30 //
//
////////////////////////////////////
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
```

```
#define Pm 2.0e-08
#define PI 3.14159265 // Pi
```

```
// ----- //
```

```
void main();
void pushKey();
```

```
double drand(); // (-1,1)の一樣乱数を与える関数
double drand01(); // (-1,1)の一樣乱数を与える関数
double normal(); // normal distri
```

```
void GenOfHA(int); // generaton of HA matrix
double Rvs(double); // if 0, -1; if -1, then 0
```

```
// ----- //
```

```
int Nq; // number of qubics
int Nb; // number of base
```

```
double a; //
```

```
double h1; //
double h2; //
```

```
double hb; //
```

```
double t[2000001]; //
double tau; //
double dt; // t[it] = (it+0.0)*dt
long it;
```

```
double L; //
```

```
double E[32]; // Energy of base
```

```

double G;                                // Gamma: coefft. for stirring Hamiltonian

double HA[32][32];                       // Pauli matrix: Stirring Hamiltonian
double HB[32][32];                       // Problem specific Hamiltonian
double H[32][32];                        // Hamiltonian

double cc[32][2000001];                  //
double cs[32][2000001];                  //

double p[32][2000001];                   //
double pttl[2000001];                    //

long NT;                                 //

FILE *fp_inp;                            // pointer of input file
FILE *fp_out;                            // pointer of output file

char InputDataFile[80];                  // input file name
char OutputDataFile[80];                 // output file name

char buf[5000];

// ----- //

void main()
{
    long i, j, k;
    int m, n;
    double tmp;

    // Input file
    sprintf(InputDataFile, "qubo1dimXHAtestGvnX_inp.dat");

    if ((fp_inp = fopen(InputDataFile, "r")) == NULL) {
        printf("Failed in Reading Input Data File! ... %s\n", InputDataFile);
        fprintf(fp_out, "Failed in Reading Input Data File! ... %s\n", InputDataFile);
        exit(1);
    }

    // Output file
    sprintf(OutputDataFile, "qubo1dimXHAtestGvnX_out.csv");

    if ((fp_out = fopen(OutputDataFile, "w")) == NULL) {
        printf("Failed in Reading Output Data File! ... %s\n", OutputDataFile);
        fprintf(fp_out, "Failed in Reading Output Data File! ... %s\n",
                OutputDataFile);
        exit(1);
    }
}

```

```

// input data from file //
fscanf(fp_inp, "%s %d", buf, &Nq);

fscanf(fp_inp, "%s %lf", buf, &a);

fscanf(fp_inp, "%s %lf", buf, &hb);
fscanf(fp_inp, "%s %lf", buf, &G);

fscanf(fp_inp, "%s %lf", buf, &dt);
fscanf(fp_inp, "%s %ld", buf, &NT);

fscanf(fp_inp, "%s %lf", buf, &L);

Nb = (int)powf(2.0, Nq+0.0);

for (m = 0; m < Nb; m++)
    fscanf(fp_inp, "%s %lf", buf, &(E[m]));

printf("Nq      = %d¥n", Nq);
printf("Nb      = %d¥n", Nb);

printf("a        = %12.6f¥n", a);

printf("hb      = %12.6f¥n", hb);
printf("G        = %12.6f¥n", G);

printf("dt      = %12.6f¥n", dt);
printf("NT      = %ld¥n", NT);

printf("L        = %12.6f¥n", L);

fprintf(fp_out, "Nq =, %d¥n", Nq);
fprintf(fp_out, "Nb =, %d¥n", Nb);

fprintf(fp_out, "a =, %12.6f¥n", a);

fprintf(fp_out, "hb =, %12.6f¥n", hb);
fprintf(fp_out, "G =, %12.6f¥n", G);

fprintf(fp_out, "dt =, %12.6f¥n", dt);
fprintf(fp_out, "NT =, %ld¥n", NT);

fprintf(fp_out, "L =, %12.6f¥n", L);
fprintf(fp_out, "¥n");

fprintf(fp_out, "n, En¥n");
for (m = 0; m < Nb; m++)
    printf("E[%d] =, %12.6f¥n", m, E[m]);
printf("¥n");

```

```

///// seach for global minimum

tau = (NT+0.0)*dt;

for (m = 0; m < Nb; m++)
    cc[m][0] = 1.0/sqrt(Nb+0.0);

for (m = 0; m < Nb; m++)
    cs[m][0] = 0.0/sqrt(Nb+0.0);

for (i = 0; i <= NT; i++)
    t[i] = (i+0.0)*dt;

for (n = 0; n < Nb; n++)
    p[n][0] = cc[n][0]*cc[n][0]+cs[n][0]*cs[n][0];
pttl[0] = 0.0;
for (n = 0; n < Nb; n++)
    pttl[0] += p[n][0];

// generate HA matrix
GenOfHA(Nq);

fprintf(fp_out, "HA[i][j]¥n");
fprintf(fp_out, "","", "");
for (j = 0; j < Nb; j++)
    fprintf(fp_out, "j = %d, ", j);
fprintf(fp_out, "¥n");
for (i = 0; i < Nb; i++) {
    fprintf(fp_out, "i = %d, ", i);
    for (j = 0; j < Nb; j++)
        fprintf(fp_out, "%12.6f, ", HA[i][j]);
    fprintf(fp_out, "¥n");
}
fprintf(fp_out, "¥n");

for (i = 0; i < Nb; i++)
    for (j = 0; j < Nb; j++)
        if (i == j)
            HB[i][j] = E[i];
        else
            HB[i][j] = 0.0;

fprintf(fp_out, "HB[i][j]¥n");
fprintf(fp_out, "","", "");
for (j = 0; j < Nb; j++)
    fprintf(fp_out, "j = %d, ", j);
fprintf(fp_out, "¥n");
for (i = 0; i < Nb; i++) {
    fprintf(fp_out, "i = %d, ", i);
    for (j = 0; j < Nb; j++)
        fprintf(fp_out, "%12.6f, ", HB[i][j]);
}

```



```

        fprintf(fp_out, "¥n");
    }
    fprintf(fp_out, "¥n");

    fprintf(fp_out, "n, En¥n");
    for (n = 0; n < Nb; n++)
        fprintf(fp_out, " %d, %12.6f¥n", n, HB[n][n]);
    fprintf(fp_out, "¥n");

    for (it = 1; it <= NT; it++) {

        for (i = 0; i < Nb; i++)
            for (j = 0; j < Nb; j++)
                H[i][j] = G*(1.0-t[it]/tau)*HA[i][j] + t[it]/tau*HB[i][j];

        for (m = 0; m < Nb; m++) {
            cc[m][it] = cc[m][it-1];
            for (n = 0; n < Nb; n++)
                cc[m][it] += 1.0/hb*H[m][n]*cs[n][it-1]*dt;
        }

        for (m = 0; m < Nb; m++) {
            cs[m][it] = cs[m][it-1];
            for (n = 0; n < Nb; n++)
                cs[m][it] -= 1.0/hb*H[m][n]*cc[n][it-1]*dt;
        }

        for (n = 0; n < Nb; n++)
            p[n][it] = cc[n][it]*cc[n][it]+cs[n][it]*cs[n][it];
        pttl[it] = 0.0;
        for (n = 0; n < Nb; n++)
            pttl[it] += p[n][it];
    }
    fprintf(fp_out, "¥n");

    fprintf(fp_out, "it, t, ");
    for (n = 0; n < Nb; n++)
        fprintf(fp_out, "cc[%d], cs[%d], ", n, n);
    for (n = 0; n < Nb; n++)
        fprintf(fp_out, "p[%d], ", n);
    fprintf(fp_out, "pttl¥n");

    for (it = 0; it <= NT; it++)
        if (it%1000 == 0) {
            fprintf(fp_out, "%ld, %12.6f, ", it, t[it]);
            for (n = 0; n < Nb; n++)
                fprintf(fp_out, "%12.6f, %12.6f, ", cc[n][it], cs[n][it]);
            for (n = 0; n < Nb; n++)
                fprintf(fp_out, "%12.6f, ", p[n][it]);
        }

```

```

        fprintf(fp_out, "%12.6f\n", pttl[it]);
    }

    pushKey();

    fclose(fp_out);
}

// ----- //

void pushKey()
{
    printf("\n      Push Return Key! ");
    getchar();
    getchar();
}

// ----- //

double drand()
{
    return 2.0*(((double)rand())/((double)RAND_MAX)-0.5);
}

// ----- //

double drand01()
{
    return (drand()+1.0)/2.0;
}

// ----- //

double normal(double x, double myu, double sgm)
{
    return 1.0/sqrt(2.0*PI*sgm*sgm)*exp(-(x-myu)*(x-myu)/(2.0*sgm*sgm));
}

// ----- //

void GenOfHA(int Nq) {

    int i, j;

    // upper left
    HA[0][0] = 0.0;
    for (j = 1; j < Nb/2; j++)
        HA[0][j] = Rvs(HA[0][j-1]);

    for (i = 1; i < Nb/2; i++)

```

```

        for (j = 0; j < Nb/2; j++)
            HA[i][j] = Rvs(HA[i-1][j]);

// lower left
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[Nb/2+i][j] = Rvs(HA[i][j]);

// upper right
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[i][Nb/2+j] = HA[Nb/2+i][j];

// lower right
for (i = 0; i < Nb/2; i++)
    for (j = 0; j < Nb/2; j++)
        HA[Nb/2+i][Nb/2+j] = HA[i][j];
}

// ----- //

double Rvs(double x) {

    if ( x == 0.0)
        return -1.0;
    else
        return 0.0;
}

// ----- //

```

入力ファイル (qubo1dimXHAtestGvnX_inp.dat)

Nq	4
a	0.0
hb	0.1
G	1.0
dt	0.00002
NT	600000
L	2.0
E0	2.0
E1	0.8
E2	0.0
E3	1.2
E4	1.5
E5	0.7
E6	0.6
E7	1.1
E8	2.0

E9	1.8
E10	1.3
E11	1.6
E12	1.7
E13	1.2
E14	0.6
E15	0.8

出力ファイル (qubo1dimXHAtestGvnX_out.csv)

第6章 量子ゲート問題

本章の量子ゲート問題は、前章の量子アニーリング問題とは大いに異なる。ただし、全く無関係と言うわけでは無くて、両者には透過性がある[6-3, 第8章]。前者は後者よりも、より抽象的、より数学的になるので、量子計算の基本概念の復習から始めよう。

古典計算の基本は1ビットと呼ばれ、0あるいは1の値をとる。物理的に0あるいは1の確定値をとるものを素子として計算機を実現する。例えば、オン/オフ状態をとるリレーのようなものが素子である。一方、量子計算の基本は1量子ビット (qubit) と呼ばれ、物理的に0および1の重ね合わせ状態にあるものを素子として計算機を実現する。例えば、極低温のジョセフソン素子のようなもので、時計回り方向の電流と反時計回り方向の電流が重ね合わせ状態になっている。

巻末のノートAとノートBが、複素ベクトルおよび量子ゲートの基礎的事項をより深く理解する上で参考になろう。

6.1 量子計算の大前提

量子計算は、以下の3か条の基本特性に基づいている。

(1) 量子状態の重ね合わせ性

1量子ビットの量子の状態 $|\psi\rangle$ は、2個の基底 $|0\rangle$ および $|1\rangle$ の重なり合った状態である：

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \quad (6.1)$$

ここで、係数 c_0 と c_1 は複素数値で、 $|c_0|^2$ と $|c_1|^2$ は観測時に基底 $|0\rangle$ と基底 $|1\rangle$ が観測される確率である。従って

$$|c_0|^2 + |c_1|^2 = 1 \quad (6.2)$$

という条件が課される。 $|\psi\rangle$ は2次元空間を表すと考えれば、ベクトル表現を用いて

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}, \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (6.3)$$

としても良い。

(2) 量子演算のユニタリー性

1量子ビットの量子の状態 $|\psi\rangle$ の変化は、 2×2 の複素数値行列 \hat{H} で表される演算子を掛けることにより引き起こされるが、この行列はユニタリーである。すなわち、転置共役行列 \hat{H}^\dagger ：

$$\hat{H}^\dagger = (\hat{H}^T)^* \quad (6.4)$$

をとると逆行列になる：

$$\hat{H}^\dagger = \hat{H}^{-1} \quad (6.5)$$

したがって

$$\hat{H}^\dagger (\hat{H} |\psi\rangle) = |\psi\rangle \quad (6.6)$$

を得る。

(3) 量子観測の確定性

計算中には量子状態そのものは重ね合わせ状態であるが、観測時には基底 $|0\rangle$ か基底 $|1\rangle$ のどちらかになる。どちらになるかは、確率 $|c_0|^2$ と確率 $|c_1|^2$ に従う。従って、どちらかが1になった状態で観測すれば、常に確率1を与える基底が観測される。すなわち、量子計算では、計算の途中では重ね合わせ状態であるが、答えを与える量子ビットの確率が1になった状態でその量子ビットを観測すれば答えが得られることになる。

6.2 量子論理ゲート

6.2.1 古典論理ゲート

図 6.1 に古典論理ゲートの NOT ゲートと NAND ゲートを示す。NOT ゲートは 1 ビット入力 1 ビット出力であり、NAND ゲートは 2 ビット入力 1 ビット出力である。

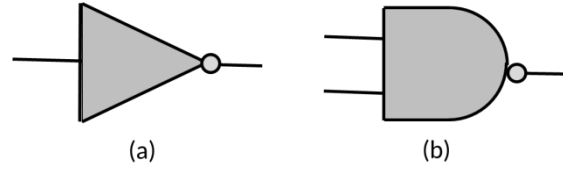


図 6.1 (a) NOT ゲート, (b) NAND ゲート

NOT ゲートの入出力は

$$\begin{aligned} 0 &\rightarrow 1, \\ 1 &\rightarrow 0 \end{aligned} \quad (6.7)$$

であり、NAND ゲートの入出力は

$$\begin{aligned} 00 &\rightarrow 1, \\ 01 &\rightarrow 1, \\ 10 &\rightarrow 1, \\ 11 &\rightarrow 0, \end{aligned} \quad (6.8)$$

である。古典ユニバーサルゲートは NAND ゲートである。

6.2.2 量子論理ゲート

古典ゲートとの違いは、入力数=出力数である。1 量子ビットの入力と出力の関係は

$$c_0|0\rangle + c_1|1\rangle \rightarrow c'_0|0\rangle + c'_1|1\rangle \quad (6.9)$$

あるいは、行列形式で

$$U \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} c'_0 \\ c'_1 \end{pmatrix} \quad (6.10)$$

と記述される。ここで、 U は 2×2 行列でユニタリーゲートである。

量子ユニバーサルゲートは制御 NOT ゲートと量子ユニタリーゲートである。制御 NOT ゲート (CNOT ゲート) は、2 ビット入力 2 ビット出力で図 6.2 で表される：

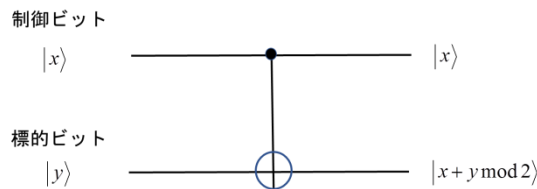


図 6.2 制御 NOT ゲート

入出力が量子ビットであることに注意して欲しい。

入力ビットの一つは制御ビットと呼ばれ、他は標的ビットと呼ばれる。図中の $x+y \bmod 2$ は 2 を法とする足し算、すなわち、足し算の結果を 2 で割った余りを意味する。具体的には、制御ビットが $|0\rangle$ のときは何もせず、 $|1\rangle$ のときは標的ビットを反転する。従って、入出力関係は

$$\begin{aligned}
|0\rangle\otimes|0\rangle &\rightarrow |0\rangle\otimes|0\rangle, \\
|0\rangle\otimes|1\rangle &\rightarrow |0\rangle\otimes|1\rangle, \\
|1\rangle\otimes|0\rangle &\rightarrow |1\rangle\otimes|1\rangle, \\
|1\rangle\otimes|1\rangle &\rightarrow |1\rangle\otimes|0\rangle
\end{aligned} \tag{6.11}$$

で与えられる。ここで、 \otimes は直積を表す。

後出の1ビットのユニタリーゲートであるパウリのXゲート σ_x ：

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{6.12}$$

は、行列表現では

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{6.13}$$

であるので

$$|0\rangle \rightarrow |1\rangle, \quad |1\rangle \rightarrow |0\rangle \tag{6.14}$$

を意味する。 $\begin{pmatrix} 1 & 0 \end{pmatrix}^T$ などの1と0を古典ビットの値と考えると、古典ゲートのNOTに相当する。しかし、量子ゲートとしての σ_x の意味は、 $\begin{pmatrix} 1 & 0 \end{pmatrix}^T$ などを1量子ビットの係数値と考えるので、古典ゲートとは全く異なる。すなわち

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_0 \end{pmatrix}: \quad c_0|0\rangle + c_1|1\rangle \rightarrow c_1|0\rangle + c_0|1\rangle \tag{6.15}$$

である。

制御NOTゲートにおいて、制御ビット入力が $c_0|0\rangle + c_1|1\rangle$ 、標的ビット入力が $c_2|0\rangle + c_3|1\rangle$ であったとする。すなわち

$$(c_0|0\rangle + c_1|1\rangle) \otimes (c_2|0\rangle + c_3|1\rangle) = c_0c_2|00\rangle + c_0c_3|01\rangle + c_1c_2|10\rangle + c_1c_3|11\rangle \tag{6.16}$$

とする。ここで

$$|xy\rangle = |x\rangle \otimes |y\rangle \tag{6.17}$$

を意味する。制御NOTゲートの入力と出力は以下ようになる：

$$c_0c_2|00\rangle + c_0c_3|01\rangle + c_1c_2|10\rangle + c_1c_3|11\rangle \rightarrow c_0c_2|00\rangle + c_0c_3|01\rangle + c_1c_2|11\rangle + c_1c_3|10\rangle \tag{6.18}$$

6.2.3 いろいろな量子論理ゲート

量子論理ゲートには、いろいろなものがある。

(1) パウリゲート

パウリのX, Y, Z スピン行列 $\sigma_x, \sigma_y, \sigma_z$ の一つ σ_x は既に(6.12)で定義した。他の σ_y, σ_z も併せて示すと

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.19}$$

である。これらの1量子ビットのユニタリーゲートにより、基底は以下のように変換される：

$$\text{パウリ X:} \quad |0\rangle \rightarrow |1\rangle, \quad |1\rangle \rightarrow |0\rangle, \tag{6.20}$$

$$\text{パウリ Y:} \quad |0\rangle \rightarrow i|1\rangle, \quad |1\rangle \rightarrow -i|0\rangle, \tag{6.21}$$

$$\text{パウリ Z:} \quad |0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow -|1\rangle \tag{6.22}$$

(2) アダマール(Hadamard)ゲート

アダマールゲート H は、図6.3に示されるように

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{6.23}$$

で定義される.

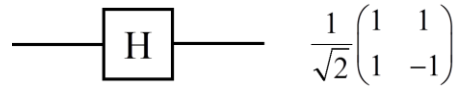


図 6.3 アダマールゲート

このゲートにより, 基底は次のように変換される:

$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \quad (6.24)$$

何故ならば

$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \quad (6.25)$$

となるからである. ここで, 極めて重要なことをしておきたい. それは $|0\rangle$ という「確定的状態」な状態にアダマールゲート H が作用すると, $1/\sqrt{2} \cdot (|0\rangle + |1\rangle)$ という「重ね合わせ状態」になる. 逆に $1/\sqrt{2} \cdot (|0\rangle + |1\rangle)$ という「重ね合わせ状態」にアダマールゲート H が作用すると, $|0\rangle$ という「確定的状態」に戻るということである.

(3) 制御ユニタリーゲート

図 6.4 に示されるように, 制御ビットにより制御されたユニタリー変換である. 制御ビットが $|0\rangle$ の時には標的ビットはそのまま通過し, $|1\rangle$ の時にはユニタリー変換を受ける. $U = \sigma_x$ のとき, 制御 NOT ゲートに等しい. すなわち, 制御ビットが $|1\rangle$ の時のみ, $|0\rangle$ を $|1\rangle$ に $|1\rangle$ を $|0\rangle$ に変換する.

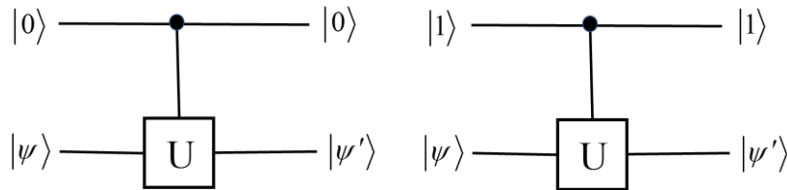


図 6.4 制御ユニタリーゲート

図 6.4 を数式で表現すると

$$\begin{aligned} |\psi\rangle &= c_0|0\rangle + c_1|1\rangle \\ |\psi'\rangle &= c'_0|0\rangle + c'_1|1\rangle \end{aligned} \quad (6.26)$$

あるいは

$$\begin{pmatrix} c'_0 \\ c'_1 \end{pmatrix} = U \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \quad (6.27)$$

と書けよう.

(4) 交換スワップゲート

二つの量子ビットを入れ替える演算は、図 6.5 で行うことができる。例えば、入力が $|\psi_1\rangle \otimes |\psi_2\rangle = |0\rangle \otimes |1\rangle$ としてみよう。最初の CNOT ゲートを通過すると、制御ビットが $|0\rangle$ だから標的ビットは素通りする。次の制御ビットは $|1\rangle$ であるので、標的ビットは $|0\rangle$ から $|1\rangle$ に変化する。最後の制御ビットは $|1\rangle$ であるので、標的ビットは $|1\rangle$ から $|0\rangle$ に変化する。すなわち、出力が $|1\rangle \otimes |0\rangle = |\psi_2\rangle \otimes |\psi_1\rangle$ になって、入力がスワップされていることが分かる。式で表せば

$$|\psi_1\rangle \otimes |\psi_2\rangle = |0\rangle \otimes |1\rangle \rightarrow |0\rangle \otimes |1\rangle \rightarrow |1\rangle \otimes |1\rangle \rightarrow |1\rangle \otimes |0\rangle = |\psi_2\rangle \otimes |\psi_1\rangle \quad (6.28)$$

となる。

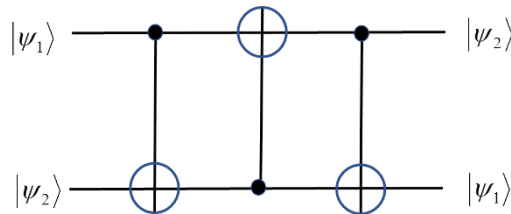


図 6.5 スワップゲート

(5) 古典（測定）ゲート

図 6.6 のゲートは、量子ビットが最終的には観測されて（読み出されて）我々が理解できる古典情報になることを示す。左側の細い線が量子ビットの入力を表し、右側の太い線が古典情報の出力を表す。

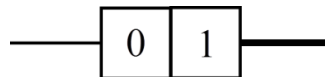


図 6.6 古典ゲート（量子情報を観測して古典情報にする）

(6) 古典制御ゲート

古典情報に基づいて計算する必要もある。これは制御ゲートの一種と考えられる。作用させる量子演算が U の場合、図 6.7 のように表す。図 6.11 に使用例がある。細線・太線がそれぞれ量子情報・古典情報を表す。

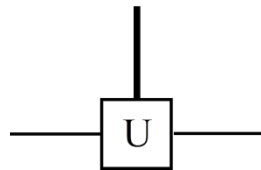


図 6.7 古典制御ユニタリーゲート（古典情報によって量子制御を行う）

量子演算を表すための簡略記法がある。計算の複雑さを避けて簡潔な表現が可能になる。 x と y は数字の 0 あるいは 1 として、アダマール変換 H を以下のように書く：

$$H|x\rangle = \frac{|0\rangle + (-1)^x |1\rangle}{\sqrt{2}} = \sum_{y=0}^1 \frac{(-1)^{xy}}{\sqrt{2}} |y\rangle \quad (6.29)$$

式 (6.29) の右辺を簡略化して

$$\sum_y \frac{(-1)^{xy}}{\sqrt{2}} |y\rangle \quad (6.30)$$

と書いても良い。

n 個の量子ビットの場合には

$$H|x_1\rangle \otimes H|x_2\rangle \otimes \cdots \otimes H|x_n\rangle = H^{(n)}|x_1 x_2 \cdots x_n\rangle = H^{(n)}|x\rangle \quad (6.31)$$

のように、0 または 1 の列 (x_1, x_2, \dots, x_n) をまとめて x と書く

$$H^{(n)}|x\rangle = \sum_{y_1} \sum_{y_2} \cdots \sum_{y_n} \frac{(-1)^{x_1 y_1 + x_2 y_2 + \cdots + x_n y_n}}{\sqrt{2^n}} |y_1 y_2 \cdots y_n\rangle = \sum_y \frac{(-1)^{xy}}{\sqrt{2^n}} |y\rangle \quad (6.32)$$

と書く。特に $x=0$ のときには

$$H^{(n)}|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \sum_y \frac{|y\rangle}{\sqrt{2^n}} \quad (6.33)$$

である。

6.2.4 絡まった状態

演算の結果、「絡まった状態」が生じる。量子ゲートが古典ゲートとは全く異なる特徴と言えよう。図 6.8 に最も簡単な例を示す。制御 NOT ビットの制御ビットに $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ を、標的ビットに $|1\rangle$ を入力してみる：

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle) = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \rightarrow \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad (6.34)$$

右辺 () 内の第 2 項が $|10\rangle$ なのは、制御ビットが $|1\rangle$ なので標的ビットが $|1\rangle$ から $|0\rangle$ になったためである ($|1\rangle \otimes |1\rangle \rightarrow |10\rangle$)。

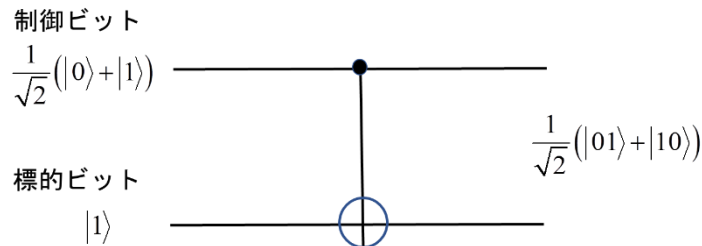


図 6.8 絡まった状態

入力側はそれぞれが他方と無関係（独立）である。一方、出力側では標的ビットは制御ビットの影響を受けている。この状態は、第 3.4 節で述べたように、いわゆる絡まった状態にあり、別々に書くことができない。このような絡まったペアを EPR ペアと呼ぶ。この図の EPR ペアの場合には、観測すると $|01\rangle$ と $|10\rangle$ がそれぞれ $(1/\sqrt{2})^2 = 0.5$ の確率で観測される。すなわち、片方が $|0\rangle$ ならば他方は $|1\rangle$ 、片方が $|1\rangle$ ならば他方は $|0\rangle$ ということになる。しかも、この関係は二つの量子ビット（量子）間の距離に関係なく成立することを意味する。これは古典力学では許容されていないことである。この点に関して、アインシュタイン (Einstein)、ポドルスキー (Podolsky)、ローゼン (Rosen) が異を唱えたので EPR ペアと呼ばれることになった。

絡まった状態を制御 NOT ビットを通すことにより、簡単に絡まっていない状態に戻せる：

$$\text{絡まった状態} : |01\rangle + |10\rangle \rightarrow |01\rangle + |11\rangle = (|0\rangle + |1\rangle) \otimes |1\rangle : \text{ほどけた状態} \quad (6.35)$$

式 (6.35) は、式 (6.34) の入力と出力を逆にしたものである

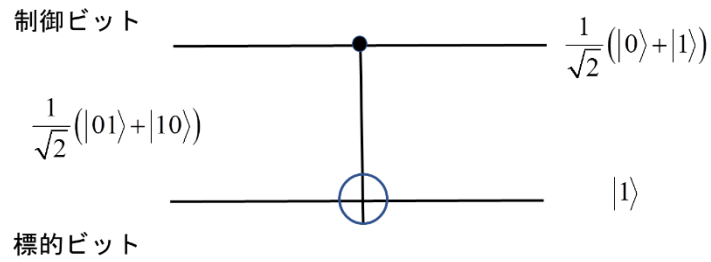


図 6.9 絡まった状態がほどける例

別の例を挙げると，図 6.10 に示されるように

$$\text{絡まった状態 : } |00\rangle + |11\rangle \rightarrow |00\rangle + |10\rangle = (|0\rangle + |1\rangle) \otimes |0\rangle : \text{ほどけた状態} \quad (6.36)$$

がある．

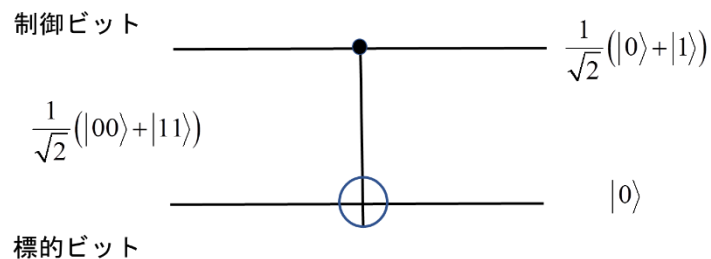


図 6.10 絡まった状態がほどける別の例

量子計算機のパワーの源は，「絡み合い」と「重ね合わせ」になる．

- (1) EPR ペアを作って，多数の並立する状態を作り出す．
- (2) 並列する絡み合い状態全体を，同時に演算することによって超並列計算を行う．
- (3) 計算の結果が含まれている絡み合いを解いて，所望の部分だけを読み出す．

6.3 量子テレポーテーション

量子計算機は簡単にいわゆるコピーを作れない．読み出した途端に波束収束で固有状態に遷移してしまう．このことは，「量子クローニングの不可能性」と呼ばれる．しかし，量子情報を遠隔地に送ることは可能である．その方法は，量子テレポーテーション（EPR ペアを使って実現）を使う．

量子情報の世界の通例で，送り手をアリス (A)，受け手をボブ (B) と呼ぶことにする．アリスが持っている 1 量子ビットの情報 $|\psi\rangle$ を，図 6.11 の量子演算回路でボブに送る．途中の遷移状態を $|\Psi_0\rangle, |\Psi_1\rangle, \dots$ と表す．

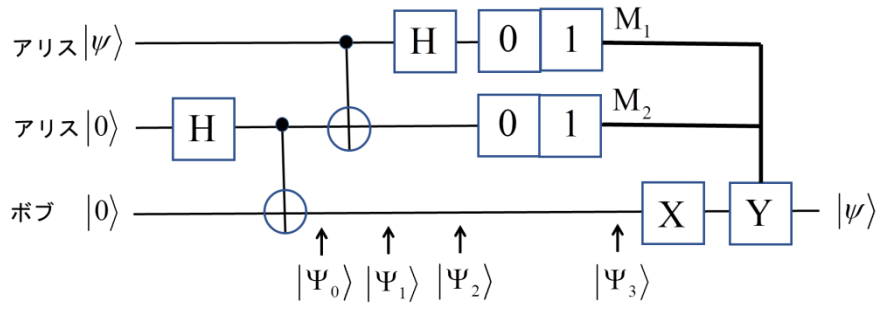


図 6.11 量子テレポーテーションを実現する量子回路

アリスは 2 量子ビットをボブは 1 量子ビットを持っている。まず

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \quad (6.37)$$

とする。

量子ビットの変化を追うと、最初の H ゲートの出力は $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ であるので、最初の CNOT ゲート通過により A の 2 番目と B の量子ビットの状態は

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

という EPR ペアになる。したがって、A の 2 個と B の 1 個の合計 3 個の量子ビットの状態 $|\Psi_0\rangle$ は

$$|\Psi_0\rangle = |\psi\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (6.38)$$

になる。

2 番目の CNOT ゲートは、A の $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ が制御情報、H ゲートを通じた $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ が標的情報であるので(アリスの 2 番目の量子ビットとボブの量子ビットは、すでに EPR ペアになっていて別々に扱えないことに注意)：

$$|\Psi_1\rangle_{\text{Before}} = |\psi\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = (c_0|0\rangle + c_1|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

であるので、CNOT ゲート通過で全体の量子ビットの状態 $|\Psi_1\rangle$ は

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}} \{ c_0|0\rangle \otimes (|00\rangle + |11\rangle) + c_1|1\rangle \otimes (|10\rangle + |01\rangle) \} \quad (6.39)$$

となる。

2 番目の H ゲートで A の一番目のビットが変化して全体の状態 $|\Psi_2\rangle$ は

$$\begin{aligned} |\Psi_2\rangle &= \frac{1}{2} \{ c_0(|0\rangle + |1\rangle) \otimes (|00\rangle + |11\rangle) + c_1(|0\rangle - |1\rangle) \otimes (|10\rangle + |01\rangle) \} \\ &= \frac{1}{2} \{ c_0(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + c_1(|010\rangle + |001\rangle - |110\rangle - |101\rangle) \} \\ &= \frac{1}{2} \{ c_0(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + c_1(|001\rangle + |010\rangle - |101\rangle - |110\rangle) \} \\ &= \frac{1}{2} \{ c_0|000\rangle + c_1|001\rangle + c_0|011\rangle + c_1|010\rangle + c_0|100\rangle - c_1|101\rangle + c_0|111\rangle - c_1|110\rangle \} \\ &= \frac{1}{2} \{ |00\rangle \otimes (c_0|0\rangle + c_1|1\rangle) + |01\rangle \otimes (c_0|1\rangle + c_1|0\rangle) + |10\rangle \otimes (c_0|0\rangle - c_1|1\rangle) + |11\rangle \otimes (c_0|1\rangle - c_1|0\rangle) \} \end{aligned} \quad (6.40)$$

となる。

$|\Psi_2\rangle$ ではアリスとボブの絡み合いは解けている。アリスは検出器 M_1 と M_2 を使って測定をする。ボブの状態は図 4.4 のパウリ X とパウリ Z ゲートを通せば、式 (6.20) と式 (6.22) によりすべて $|\psi\rangle$ になる。何故なら、式 (6.40) の最右辺第 1 項はアリスの 2 個のビットが $|00\rangle$ のため $M_1 = M_2 = 0$ であるので、ボブの $c_0|0\rangle + c_1|1\rangle = |\psi\rangle$ はそのまま通過する。第 2 項は $M_1 = 0$, $M_2 = 1$ であるので、 X ゲートのみが有効になり

$$X(c_0|1\rangle + c_1|0\rangle) = c_0|0\rangle + c_1|1\rangle = |\psi\rangle \quad (6.41)$$

となる。第 3 項は $M_1 = 1$, $M_2 = 0$ であるので、 Z ゲートのみが有効になり

$$Z(c_0|0\rangle - c_1|1\rangle) = c_0|0\rangle + c_1|1\rangle = |\psi\rangle \quad (6.42)$$

となる。第 4 項は $M_1 = 1$, $M_2 = 1$ であるので、 X ゲートと Z ゲートの両方が有効になり

$$ZX(c_0|1\rangle - c_1|0\rangle) = Z(c_0|0\rangle - c_1|1\rangle) = c_0|0\rangle + c_1|1\rangle = |\psi\rangle \quad (6.43)$$

となる。各項の係数は $1/2$ であるので、各項が同じ確率 $1/4$ で起き、その時ボブのビットはどの項が起きてても $|\psi\rangle$ である。故に、情報をアリスからボブにテレポーテーションできたことになる。

6.4 ドイッチージョサ (Deutsch-Jozsa) のアルゴリズム

2 量子ビット $|x\rangle \otimes |y\rangle$ を $|x\rangle \otimes |y \oplus f(x)\rangle$ に変換する操作を考える。 \oplus は 2 を法とする和である。すなわち、 $0 \oplus 0 = 1 \oplus 1 = 0$, $0 \oplus 1 = 1 \oplus 0 = 1$ である。混乱を避けるために、2 量子ビット $|x\rangle \otimes |y\rangle = |xy\rangle$ を $|x, y\rangle$ と書くことにする。すなわち、 $|x, y\rangle$ を $|x, y \oplus f(x)\rangle$ とする。ここで、 $f(x)$ は 0 か 1 の値を取る任意の関数である。このような操作をするゲートを図 6.12 に示す。図中の U はこの操作がユニタリー変換であることを強調するためである。図中の x , y , $f(x)$ は適宜 $|x\rangle$, $|y\rangle$, $|f(x)\rangle$ と考えて欲しい。誤解を招き易いのは第 2 量子ビットである。第 1 量子ビットの入力は $|x\rangle$ で出力も $|x\rangle$ であるのに対して、第 2 量子ビットの入力は $|y\rangle$ で出力は $|y \oplus f(x)\rangle$ である。

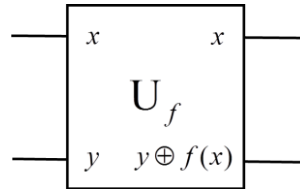


図 6.12 一般的な 2 量子ビット操作をするゲート

このようなゲートは制御 NOT ゲートと同様に、一般に絡み合った状態を産み出す。以下において、ドイッチのアルゴリズムを解説する。ドイッチのアルゴリズムは量子計算の本質を始めて示したもので、古典計算では 2 回やらないといけない計算が、量子計算では 1 回で済むこと、すなわち計算量が $1/2$ になることを示したものである。

6.4.1 ドイッチのアルゴリズム

図 6.12 のゲートの前後に、3 個のアダマールゲートを付け加えたドイッチのゲートと呼ばれるものを図 6.13 に示す。

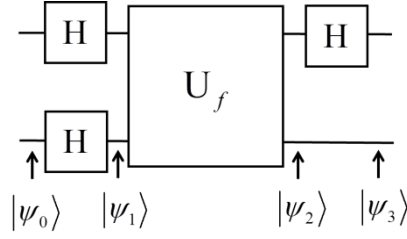


図 6.13 ドイッチのゲート

このゲートの第1ビットに $|0\rangle$ 、第2ビットに $|1\rangle$ を入れてその変位を追及すると以下ようになる：

$$|\psi_0\rangle = |01\rangle \quad (6.44)$$

$$|\psi_1\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) \quad (6.45)$$

$$|\psi_2\rangle = \begin{cases} \pm \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) & \text{when } f(0) = f(1) \\ \pm \frac{1}{2}(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle) & \text{when } f(0) \neq f(1) \end{cases} \quad (6.46)$$

式(6.44)と式(6.45)は説明の必要はないであろう。式(6.46)は説明が必要である。式(6.45)を書き直すと

$$2|\psi_1\rangle = (|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) = |0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle \quad (6.47)$$

となるので

$$2|\psi_2\rangle = |0\rangle \otimes |0 \oplus f(0)\rangle - |0\rangle \otimes |1 \oplus f(0)\rangle + |1\rangle \otimes |0 \oplus f(1)\rangle - |1\rangle \otimes |1 \oplus f(1)\rangle \quad (6.48)$$

となる。 $f(0) = f(1) = 0$ のときには

$$2|\psi_2\rangle = |0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle = +(|0\rangle + |1\rangle) \otimes |0\rangle - (|0\rangle + |1\rangle) \otimes |1\rangle = +(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) \quad (6.49)$$

となり、 $f(0) = f(1) = 1$ のときには

$$2|\psi_2\rangle = |0\rangle \otimes |1\rangle - |0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle = -(|0\rangle + |1\rangle) \otimes |0\rangle + (|0\rangle + |1\rangle) \otimes |1\rangle = -(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) \quad (6.50)$$

となり、 $f(0) = 0, f(1) = 1$ のときには

$$2|\psi_2\rangle = |0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle = +(|0\rangle - |1\rangle) \otimes |0\rangle - (|0\rangle - |1\rangle) \otimes |1\rangle = +(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle) \quad (6.51)$$

となり、 $f(0) = 1, f(1) = 0$ のときには

$$2|\psi_2\rangle = |0\rangle \otimes |1\rangle - |0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle = -(|0\rangle - |1\rangle) \otimes |0\rangle + (|0\rangle + |1\rangle) \otimes |1\rangle = -(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle) \quad (6.52)$$

となるので、式(6.46)が正しいと言える。

さらに、アダマールゲートはユニタリーであるので、任意の $|\psi\rangle$ に対して

$$H(H|\psi\rangle) = H^\dagger(H|\psi\rangle) = H^{-1}(H|\psi\rangle) = |\psi\rangle \quad (6.53)$$

であるので

$$|\psi_3\rangle = \begin{cases} \pm \frac{1}{\sqrt{2}}|0\rangle \otimes (|0\rangle - |1\rangle) & \text{when } f(0) = f(1) \\ \pm \frac{1}{\sqrt{2}}|1\rangle \otimes (|0\rangle - |1\rangle) & \text{when } f(0) \neq f(1) \end{cases} = \pm |f(0) \oplus f(1)\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (6.54)$$

となる。要するに、関数 $f(x)$ について $f(0)$ と $f(1)$ の和 $f(0) \oplus f(1)$ を一度に計算している。古典計算機ならば、まず $f(0)$ と $f(1)$ を計算し次にその和 $f(0) \oplus f(1)$ を計算すると言うように、別々に計算しなければならない所である。2度の計算が1度の計算になっている。

式(6.46)を書き直せば

$$|\psi_2\rangle = \pm \sum_{x=0}^1 \frac{(-1)^{f(x)}}{\sqrt{2}} |x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \quad (6.55)$$

となることに注意しておく。

6.4.2 ドイッチージョサのアルゴリズム

図 6.12 を一般化した図 6.14 に与えられる操作を考えよう。入力が 1 量子ビットから n 量子ビットに拡張されている。 n 量子ビットの入力 $x = (x_1, x_2, \dots, x_n)$ に対して、1 量子ビットの関数 $f(x) = f(x_1, x_2, \dots, x_n)$ が定義されているとする。誤解を招かないように言えば、 $|x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle \otimes |y\rangle$ の入力に対して $|x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle \otimes |y \oplus f(x_1, x_2, \dots, x_n)\rangle$ が出力される。また、 $x = (x_1, x_2, \dots, x_n)$ の総数は 2^n になる。 n が大きくなるとともに爆発的に大きくなる。

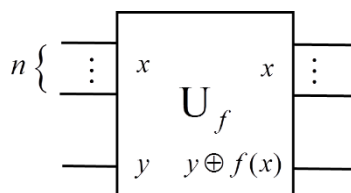


図 6.14 入力が $(n+1)$ 量子ビットのドイッチのゲート

この関数 $f(x)$ には以下のような定置型と分布型という二つの性質があるものとする。

定置型： $f(x) =$ すべての x について 0 または 1

分布型： $f(x) =$ 半分の x について 0, 残りの半分について 1

関数 $f(x)$ が定置型か分布型かを判別する問題を考えて見よう。

- (1) アリス (A) からボブ (B) に、 n 量子ビットの x と 1 ビットの y を渡す。
- (2) B はこの x と y を図 6.14 の量子計算機に掛ける。
- (3) A はその結果を B から受け取り $f(x)$ がどちらの型かを判別する。

n 量子ビット x の総数は 2^n なので、その半分は $2^n/2 = 2^{n-1}$ である。何も知らずにこのゲームに勝とうとすると、最悪の場合 $2^n/2 + 1$ 個の n 量子ビット x を試すことになる。何故なら、たまたま $2^n/2$ の $f(x)$ が同じであったとしても、それを以て $f(x)$ が定置型か分布型かは判別できない。巡回セールスマン問題は、すべての経路を調べないと正解が分からないので、同様な問題と言えよう。このような問題を組み合わせ爆発問題という。

ところが、量子計算の場合には、一組の x と y が分かれば良いと言う驚くべきことになる。これを実現する方法が図 6.15 である。A は B に任意の x と y ではなく、以下に示されるような x と y を意図的に渡す：

$$|\psi_0\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle \otimes |y\rangle = |0\rangle \otimes |0\rangle \otimes \dots \otimes |0\rangle \otimes |y\rangle = |00\dots 0\rangle \otimes |1\rangle \quad (6.56)$$

これにアダマール変換を施すと、式 (6.45) より

$$\begin{aligned} |\psi_1\rangle &= \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \dots \otimes (|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} (|00\dots 0\rangle + |00\dots 1\rangle + \dots + |11\dots 1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\ &= \sum_{(x_1, x_2, \dots, x_n) \in \{0,1\}^n} \frac{|x_1 x_2 \dots x_n\rangle}{\sqrt{2^n}} \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = \sum_{x \in \{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}} \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \end{aligned} \quad (6.57)$$

となる.

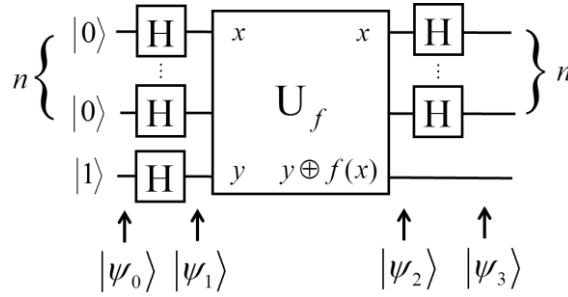


図 6.15 ドイッチージョサのアルゴリズム

従って, 式 (6.55) により

$$|\psi_2\rangle = \sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)} |x\rangle}{\sqrt{2^n}} \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \quad (6.57)$$

を得る.

ここで, 1 量子ビット $|x\rangle$ に対するアダマールゲートの作用が

$$H|x\rangle = \sum_{z=0,1} \frac{(-1)^{xz}}{\sqrt{2}} |z\rangle \quad (6.58)$$

であり, $|n\rangle$ 量子ビットに作用させると

$$H^n |x\rangle = H^n |x_1 x_2 \cdots x_n\rangle = \sum_{z_1, z_2, \dots, z_n} \frac{(-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n}}{\sqrt{2^n}} |z_1 z_2 \cdots z_n\rangle = \sum_z \frac{(-1)^{xz}}{\sqrt{2^n}} |z\rangle \quad (6.59)$$

となる. 故に, 式 (6.54) より

$$|\psi_3\rangle = \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{(-1)^{xz + f(x)}}{\sqrt{2^n}} |z\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \quad (6.60)$$

となる.

ここで, $|z\rangle = |00 \cdots 0\rangle$ の項に注目すると $xz = 0$ なので, その係数は

$$\text{Coefft. of } |z\rangle = |00 \cdots 0\rangle \text{ in } \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{(-1)^{xz + f(x)}}{2^n} |z\rangle = \sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)}}{2^n} \quad (6.61)$$

と求まる. 従って

- ・ $f(x)$ が定置型ならば, 分子のすべて 2^n 個が +1 ($f(x) = 0$ のとき) または -1 ($f(x) = 1$ のとき) だから, 係数は ± 1 になる.
- ・ $f(x)$ が分布型ならば, 分子の半数 $2^n/2$ 個づつが +1 ($f(x) = 0$ のとき) と -1 ($f(x) = 1$ のとき) で総和は 0 だから, 係数は 0 になる.

故に, この時点で観測すれば, 定置型ならば $|z\rangle = |00 \cdots 0\rangle$ を観測する確率は $(\pm 1)^2 = 1$ であるので必ず $|z\rangle = |00 \cdots 0\rangle$ が観測されるのに対し, 分布型の場合にはその確率は $(0)^2 = 0$ となるので $|z\rangle = |00 \cdots 0\rangle$ 以外のものが観測される. 要するに, 量子計算機ならば, 1 度の計算で $f(x)$ が定置型か分布型か判別できることになる. 古典計算機ならば, 最悪の場合 $2^n/2 + 1$ 回の試行が必要な問題である.

ドイッチのアルゴリズムにどんな応用があるのかは分からないが、量子計算のパワーを垣間見せてくれるという意味合いで歴史的に貴重なものと言えよう。この研究を通して量子計算の数理的な基礎が築かれた訳で偉業と言ってよいであろう。

6.5 ショアのアルゴリズム

ドイッチの偉業に並ぶもう一つの偉業がショアのアルゴリズムであろう。ショアの大きな業績により、量子計算機の研究が一気に加速されたようである。その理由は、現在の暗号方式は公開鍵暗号方式と呼ばれるが、大きな素数の素因数分解には、スパコンを用いても膨大な時間が掛かることがベースになっている。しかし、量子計算ならショアのアルゴリズムにより簡単に素因数分解できることが示されたからである。

6.5.1 公開鍵暗号

現在の暗号方式は公開鍵暗号方式と呼ばれる。以前の共通鍵暗号方式では、送信者と受信者の両者がただ1つの共通の鍵を用いるために暗号の秘匿に問題があった。そこで**公開鍵暗号**が考案された。

大きな数の素因数分解はスパコンでも難しいことを利用する。公開鍵暗号とは図 6.16 のようにアリス(送信者)からボブ(受信者)への情報通信をボブが作った施錠鍵を用い、第3者に読み取られないように行う。

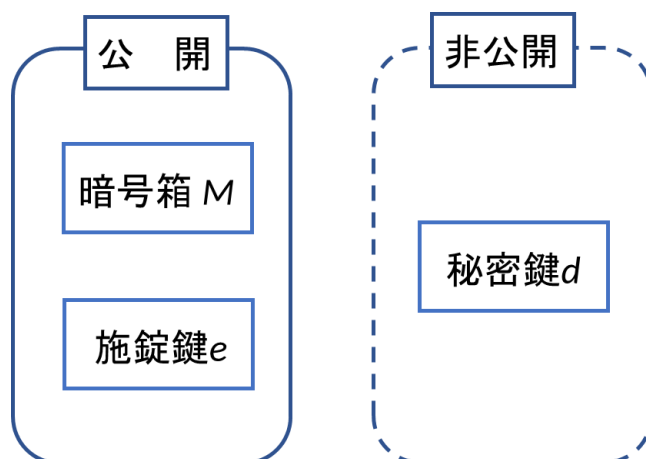


図 6.16 アリスは、公開されているボブが作った施錠鍵 e と箱 M を用いて、送りたい秘密の情報を箱の中に入れ施錠してボブへ送る。ボブは自分だけが持っている秘密鍵 d を用いて解錠し、アリスからの秘密の情報を得る。

暗号作成と復号の手順を示そう。

- (1) ボブは施錠鍵 e と箱 M 、解錠鍵(秘密鍵) d をつぎのように作成し、 e と M だけ公開する。
 - (a) 二つの大きな素数 p と q を適当に選択し、 $M = pq$ を計算する。
 - (b) $L = (p-1)(q-1)$ を計算し、 L と互いに素で L より小さい奇数 e を適当に定める。
 - (c) ed を L で割った余りが 1 となる d を探す。式で書くと $ed \bmod L = 1$ である。
- (2) アリスはメッセージを整数で表した P から暗号文 C を $C = P^e \bmod M$ により計算する。ただし、 $P < M$ でなければならない。

暗号文作成の例を以下に示す。送りたい言葉を LOVE としよう。2進法での ASCII コードは

$$[L]=1001100_{(2)} \quad [O]=1001111_{(2)} \quad [V]=1010110_{(2)} \quad [E]=1000101_{(2)} \quad (6.62)$$

である。これより、アリスは原文：

$$P = 1001100100111110101101000101_{(2)} = 160688965_{(10)} \quad (6.63)$$

を作成する。次にアリスは暗号文：

$$C = P^e \bmod M = 160688965^e \bmod M \quad (6.64)$$

を作成する。

(3) ボブは受け取った暗号文 C から元のメッセージ：

$$P = C^d \bmod M \quad (6.65)$$

を解読する。

式 (6.65) の証明には、フェルマーの小定理：

$$\text{「} g \text{ が素数 } p \text{ の倍数でないとき, } g^{p-1} \bmod p = 1 \text{ が成り立つ」} \quad (6.66)$$

が必要である。 $p=5$, $g=1, 2, 3$ としてフェルマーの小定理が成り立つ例を挙げる：

$$1^4 \bmod 5 = 1, \quad 2^4 \bmod 5 = 16 \bmod 5 = 1, \quad 3^4 \bmod 5 = 81 \bmod 5 = 1$$

式 (6.65) の証明を以下に示す：

$$\begin{aligned} C^d \bmod M &= (P^e \bmod M)^d \bmod M \\ &= \left[(P^e)^d \bmod M \right] \bmod M \\ &\quad \text{because } (P^e)^d \bmod M = (\lambda M + \alpha)^d \bmod M = \alpha^d \bmod M \\ &= (P^e \bmod M)^d \bmod M \\ &= P^{ed} \bmod M \\ &= P^{k(p-1)(q-1)+1} \bmod M \\ &= P \cdot P^{k(p-1)(q-1)} \bmod M \\ &= P \cdot (\gamma pq + 1) \bmod M \\ &\quad \text{because of Fermat's little theorem} \\ &= (\gamma PM + P) \bmod M = P \\ &\quad \text{because } P < M \quad \text{Q.E.D.} \end{aligned} \quad (6.67)$$

簡単な例で、暗号の作成と復号を行ってみよう。

(1) ボブは施錠鍵 $e=7$ と箱 $M=33$, 解錠鍵 (秘密鍵) $d=3$ を作成し $e=7$ と $M=33$ だけを公開する。

(a) 二つの大きな素数 $p=3$ と $q=11$ を選択し, $M=pq=33$ を計算する。

$L=(p-1)(q-1)=20$ を計算し, $L=20$ と互いに素 (共通の約数を持たない) で $L=20$ より小さい奇数として $e=7$ を選ぶ。

(b) $ed=7d$ を $L=20$ で割ると余りが 1 となる数 d を探す。ここでは $d=3$ とする。

(2) アリスはメッセージを整数で表した $P=5$ から暗号文 C を $C=P^e \bmod M$ より計算する。

$$C = P^e \bmod M = 5^7 \bmod 33 = 78125 \bmod 33 = 14 \quad (6.68)$$

(3) ボブは受け取った暗号文 C から元メッセージを $P=C^d \bmod M$ で計算する。

$$P = C^d \bmod M = 14^3 \bmod 33 = 2744 \bmod 33 = 5 \quad (6.69)$$

もしも, M を素因数分解できて p と q が分かってしまうと, $L=(p-1)(q-1)$ が分かり e は公開されているので, 何回かの試行で $ed \bmod L = 1$ を d について解くことが可能となり, 暗号がばれてしまう。 M が大きい場合には, M の素因数分解は現在の計算機の能力では, 膨大な時間が掛かってしまい実質的に不可能であり, 暗号が解読されないことになる。

量子コンピューターを用いるショアのアルゴリズムでは、素因数分解を高速に解けることになるのでその衝撃は大きかった。

6.5.2 量子計算によるショアのアルゴリズムを導くための準備（素因数分解）

以下の議論を理解する助けとして、数 M を素因数 p, q に分解するアルゴリズムについて述べる。以下のようなアルゴリズムがある。

- (1) 最初に M よりも小さく、お互いに約数を持たない次のような奇数 e を選ぶ：

$$e \in 1, 2, \dots, M-1, \quad \gcd(e, M) = 1 \quad (6.70)$$

ここで、 $\gcd(e, M)$ は e と M の最大公約数を表す。

- (2) T を整数として

$$M = e^T - 1 \quad (6.71)$$

ならば

$$f(x+T) = e^{T+x} \bmod M = (M+1)e^x \bmod M = (Me^x + e^x) \bmod M = e^x \bmod M = f(x) \quad (6.72)$$

であるので関数 $f(x) = e^x \bmod M$ は周期 T を有す。すなわち、 $M = \text{given}$ であるから T は e の関数である。この性質を使うと、2 個の素数の積である整数 M が与えられたとき、周期 T が偶数なら以下の (3) の手順で、 M を素因数分解できる。周期 T が奇数なら、(1) に戻り e を選び直し、 T が偶数になるまで繰り返す。

簡単な例を示す。 $M = 15$ および $e = 2$ とすると

$$\begin{aligned} e^x \bmod M &= 2^x \bmod 15 \\ &= 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, \dots \bmod 15 \\ &= 1, 2, 4, 8, 16, 32, 64, 128, \dots = 1, 2, 4, 8, 1, 2, 4, 8, \dots \end{aligned} \quad (6.73)$$

であるので、 $T = 4$ が求まる。

- (3) 周期性条件より

$$e^x \bmod M = e^{x+T} \bmod M \quad (6.74)$$

となる。従って

$$e^T \bmod M = 1 \quad (6.75)$$

を得る。式 (6.75) を満たす T は位数と呼ばれる。 T が最小の周期すれば、式 (6.71) が言える。従って

$$e^T - 1 = (e^{T/2} + 1)(e^{T/2} - 1) \bmod M = 0 \quad (6.76)$$

すなわち、 $e^{T/2} + 1$ と $e^{T/2} - 1$ は、 M より小さければ M の公約数である。

- (4) 求めた最大公約数が p, q であるならば、 M の素因数分解ができたことになる。

素因数分解アルゴリズムが位数発見アルゴリズムとなることは、周期 T が式 (6.75) を満たすので、位数になっているためである。

(1) から (4) までの事項を簡単な数値例で確認しよう。

【数値例 1】 $M = 6$ の場合

$1 < e < 6$ の整数で M と共通の約数を持たない奇数 $e = 5$ を選ぶ。 $f(x) = e^x \bmod M = 5^x \bmod 6$ の周期 T は

$$5^0 \bmod 6 = 1, \quad 5^1 \bmod 6 = 5, \quad 5^2 \bmod 6 = 1, \quad 5^3 \bmod 6 = 5, \quad 5^4 \bmod 6 = 1, \dots \quad (6.77)$$

なので $T = 2$ (偶数) となり

$$\gcd(e^{T/2} \pm 1, M) = \gcd(5^1 \pm 1, 6) = \gcd(5 \pm 1, 6) = 6, 2 \quad (6.78)$$

であるので、素因数 2 が見つかった。他は $M/2 = 6/2 = 3$ となる。

【数値例 2】 $M = 15$ の場合

$1 < e < 15$ の整数で M と共通の約数を持たない奇数 $e = 7$ を選ぶ。 $f(x) = e^x \bmod M = 7^x \bmod 15$ の周期 T は

$$7^0 \bmod 15 = 1, \quad 7^1 \bmod 15 = 7, \quad 7^2 \bmod 15 = 4, \quad 7^3 \bmod 15 = 13, \quad 7^4 \bmod 15 = 1, \dots \quad (6.79)$$

なので $T = 4$ (偶数) となり

$$\gcd(e^{T/2} \pm 1, M) = \gcd(7^2 \pm 1, 15) = \gcd(49 \pm 1, 15) = 5, 3 \quad (6.80)$$

であるので、素因数 3 と 5 の二つとも見つかった。

【数値例 3-1】 $M = 21$ で、 $e = 5$ の場合

$1 < e < 21$ の整数で M と共通の約数を持たない奇数 $e = 5$ を選ぶ。 $f(x) = e^x \bmod M = 5^x \bmod 21$ の周期 T は

$$\begin{aligned} 5^0 \bmod 21 &= 1, & 5^1 \bmod 21 &= 5, & 5^2 \bmod 21 &= 4, & 5^3 \bmod 21 &= 20, \\ 5^4 \bmod 21 &= 16, & 5^5 \bmod 21 &= 17, & 5^6 \bmod 21 &= 1, \dots \end{aligned} \quad (6.81)$$

なので $T = 6$ (偶数) となり

$$\gcd(e^{T/2} \pm 1, M) = \gcd(5^3 \pm 1, 21) = \gcd(125 \pm 1, 21) = 7 \text{ from } 126 \quad (6.82)$$

であるので、素因数 7 が見つかった。他は $M/7 = 21/7 = 3$ となる。

【数値例 3-2】 $M = 21$ で、 $e = 13$ の場合

$1 < e < 21$ の整数で M と共通の約数を持たない奇数 $e = 13$ を選ぶ。 $f(x) = e^x \bmod M = 13^x \bmod 21$ の周期 T は

$$13^0 \bmod 21 = 1, \quad 13^1 \bmod 21 = 13, \quad 13^2 \bmod 21 = 7, \quad 13^3 \bmod 21 = 13, \quad 13^4 \bmod 21 = 1, \dots \quad (6.83)$$

なので $T = 4$ (偶数) となり

$$\gcd(e^{T/2} \pm 1, M) = \gcd(13^2 \pm 1, 21) = \gcd(169 \pm 1, 21) = 7 \text{ from } 168 \quad (6.84)$$

であるので、素因数 7 が見つかった。他は $M/7 = 21/7 = 3$ となる。

【数値例 3-3】 $M = 21$ で、 $e = 17$ の場合

$1 < e < 21$ の整数で M と共通の約数を持たない奇数 $e = 17$ を選ぶ。 $f(x) = e^x \bmod M = 17^x \bmod 21$ の周期 T は

$$\begin{aligned} 17^0 \bmod 21 &= 1, & 17^1 \bmod 21 &= 17, & 17^2 \bmod 21 &= 16, & 17^3 \bmod 21 &= 20, & 17^4 \bmod 21 &= 4, \\ 17^5 \bmod 21 &= 5, & 17^6 \bmod 21 &= 1, \dots \end{aligned} \quad (6.85)$$

なので $T = 6$ (偶数) となり

$$\gcd(e^{T/2} \pm 1, M) = \gcd(17^3 \pm 1, 21) = \gcd(4913 \pm 1, 21) = 3, 7 \text{ from } 4914 \quad (6.86)$$

であるので、素因数 3 と 7 が見つかった。

6.5.3 量子離散的フーリエ変換

量子計算による暗号解読は、量子計算特有の機能に基づく高速計算が用いられる。高速計算のキーとなる機能とは、量子アニーリング問題の場合も量子ゲート問題の場合も基本的には同じで

- ・重ね合わせによる全数チェック、すなわち超並列計算。
- ・有効な状態の確率を高くする絞りこみ。
- ・測定による波束の収束。

であろう。アニーリング問題の場合とゲート問題の場合の違いは、絞り込み方法の違いと思われる。

アニーリングの場合には、あらゆる状態が均等に存在する高温状態をエネルギー最小の状態に移行させることにより、有効な状態の確率を大きくする絞りこみを行い測定する。これは極めて分かり易い。

一方、ゲート問題の場合には、いろいろな絞り込み方法があると思うが、量子計算による暗号解読のキ

一である位数計算問題では、有効な状態に内在する周期性を利用して、検討範囲を狭い一周内に絞りこむ。これは極めて分かり難い。

量子計算による暗号解読では、以下のようなアルゴリズムが用いられる：

- ・量子離散的フーリエ変換
- ・位相推定問題
- ・位数計算

が必要となる。まず、量子離散的フーリエ変換の説明をする。

(1) 量子離散的フーリエ変換の定義

ショアのアルゴリズムの核心は位数計算であるが、量子離散的フーリエ変換が使われている。無限区間 $-\infty \leq t < \infty$ における連続関数のフーリエ変換は次式により与えられる：

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt \quad (6.87)$$

逆変換は

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{-i\omega t} d\omega \quad (6.88)$$

である。

有限区間 $0 \leq t < NT$ における離散的時間：

$$t_j = jT, \quad j = 0, 1, \dots, N-1 \quad (6.89)$$

における数列（状態）：

$$|x\rangle = (x_0, x_1, \dots, x_{N-1})^T \quad (6.90)$$

から、数列（状態）：

$$|y\rangle = (y_0, y_1, \dots, y_{N-1})^T \quad (6.91)$$

への離散的なフーリエ変換は

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{i2\pi jk/(NT)} \quad (6.92)$$

で定義される。便宜的に $T=1$ とすると

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{i2\pi jk/N} \quad (6.93)$$

となる。

「フーリエ変換は、座標変換と同様な表現の変換である」という捉え方をすれば、「一つの状態のある基底系 $|j\rangle$, $j=0,1,\dots,N-1$ での表現を、他の基底系 $|k\rangle$, $k=0,1,\dots,N-1$ での表現に変換する行為」と考えられよう。基底が変わっても状態は変わらないので、式 (6.92) より

$$\sum_{j=0}^{N-1} x_j |j\rangle = \sum_{k=0}^{N-1} y_k |k\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{i2\pi jk/N} |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \sum_{k=0}^{N-1} e^{i2\pi jk/N} |k\rangle \quad (6.94)$$

が成り立つ。この最左辺と最右辺より、基底 $|k\rangle$ から基底 $|j\rangle$ への変換は

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi jk/N} |k\rangle \quad (6.95)$$

で与えられることになる。これを量子離散的フーリエ変換と呼ぶ。

$N=2$ の場合、式 (6.95) より

$$\begin{aligned}
|0\rangle_j &= \frac{1}{\sqrt{2}} \sum_{k=0}^1 e^{i2\pi \cdot 0 \cdot k/2} |k\rangle = \frac{1}{\sqrt{2}} (|0\rangle_k + |1\rangle_k) \\
|1\rangle_j &= \frac{1}{\sqrt{2}} \sum_{k=0}^1 e^{i2\pi \cdot 1 \cdot k/2} |k\rangle = \frac{1}{\sqrt{2}} (|0\rangle_k - |1\rangle_k)
\end{aligned} \tag{6.96}$$

となる。これは、式(6.24)で示されたアダマールゲートである。2次元の量子離散的フーリエ変換はアダマールゲートに他ならない。式(6.95)により基底が変わるので観測値が変わることになる。

$N=4$ の場合は

$$\begin{aligned}
|0\rangle_j &= \frac{1}{\sqrt{4}} \sum_{k=0}^3 e^{i2\pi \cdot 0 \cdot k/4} |k\rangle = \frac{1}{2} (|0\rangle_k + |1\rangle_k + |2\rangle_k + |3\rangle_k) \\
|1\rangle_j &= \frac{1}{\sqrt{4}} \sum_{k=0}^3 e^{i2\pi \cdot 1 \cdot k/4} |k\rangle = \frac{1}{2} (|0\rangle_k + i|1\rangle_k - |2\rangle_k - i|3\rangle_k) \\
|2\rangle_j &= \frac{1}{\sqrt{4}} \sum_{k=0}^3 e^{i2\pi \cdot 2 \cdot k/4} |k\rangle = \frac{1}{2} (|0\rangle_k - |1\rangle_k + |2\rangle_k - |3\rangle_k) \\
|3\rangle_j &= \frac{1}{\sqrt{4}} \sum_{k=0}^3 e^{i2\pi \cdot 3 \cdot k/4} |k\rangle = \frac{1}{2} (|0\rangle_k - i|1\rangle_k - |2\rangle_k + i|3\rangle_k)
\end{aligned} \tag{6.97}$$

となる。この変換はユニタリー変換である。行列表現を使うと見通しがいい。上式の変換は基底 $|j\rangle$ の基底 $|k\rangle$ での表現であり

$$\begin{pmatrix} |0\rangle_j \\ |1\rangle_j \\ |2\rangle_j \\ |3\rangle_j \end{pmatrix} = \Gamma \begin{pmatrix} |0\rangle_k \\ |1\rangle_k \\ |2\rangle_k \\ |3\rangle_k \end{pmatrix} \tag{6.98}$$

と考えると

$$\Gamma = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \tag{6.99}$$

と書ける。 Γ の転置共役を Γ^\dagger とすると、ユニタリー性：

$$\Gamma^\dagger \Gamma = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} \times \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.100}$$

を満たすが、 $\Gamma^\dagger = \Gamma$ を満たさないで、エルミートではない。

(2) 量子離散的フーリエ変換の量子回路

量子離散的フーリエ変換を量子コンピューターで行うためには、 $|j\rangle, |k\rangle, (j, k=0,1,\dots,N)$ を量子ビットで表す必要がある。 $N=2^n$ として10進数である j と k を2進数：

$$\begin{aligned}
j &= j_1 \cdot 2^{n-1} + j_2 \cdot 2^{n-2} + \dots + j_n \cdot 2^0 \\
k &= k_1 \cdot 2^{n-1} + k_2 \cdot 2^{n-2} + \dots + k_n \cdot 2^0
\end{aligned} \tag{6.101}$$

で表し、基底ベクトル $|j\rangle$ と $|k\rangle$ を

$$\begin{aligned} |j\rangle &= |j_1 j_2 \cdots j_n\rangle = |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_n\rangle \\ |k\rangle &= |k_1 k_2 \cdots k_n\rangle = |k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_n\rangle \end{aligned} \quad (6.102)$$

とし、1より小さい数も2進数で表す：

$$\begin{aligned} 0.j_1 j_2 \cdots j_m &= \frac{j_1}{2^1} + \frac{j_2}{2^2} + \cdots + \frac{j_m}{2^m} \\ 0.k_1 k_2 \cdots k_m &= \frac{k_1}{2^1} + \frac{k_2}{2^2} + \cdots + \frac{k_m}{2^m} \end{aligned} \quad (6.103)$$

量子離散的フーリエ変換の表現には、基底ベクトルに対する動作（変換）を表す式(6.95)を量子ビットで表せばよい：

$$\begin{aligned} |j\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi jk/2^n} |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \cdots \sum_{k_n=0}^1 e^{i2\pi j(k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0)/2^n} |k_1 k_2 \cdots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \cdots \sum_{k_n=0}^1 e^{i2\pi j\left(\frac{k_1}{2^1} + \frac{k_2}{2^2} + \cdots + \frac{k_n}{2^n}\right)} |k_1 k_2 \cdots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \cdots \sum_{k_n=0}^1 e^{i2\pi j \frac{k_1}{2^1}} e^{i2\pi j \frac{k_2}{2^2}} \cdots e^{i2\pi j \frac{k_n}{2^n}} |k_1 k_2 \cdots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \sum_{k_2=0}^1 \cdots \sum_{k_n=0}^1 e^{i2\pi j \frac{k_1}{2^1}} |k_1\rangle \otimes e^{i2\pi j \frac{k_2}{2^2}} |k_2\rangle \otimes \cdots \otimes e^{i2\pi j \frac{k_n}{2^n}} |k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 e^{i2\pi j \frac{k_1}{2^1}} |k_1\rangle \otimes \sum_{k_2=0}^1 e^{i2\pi j \frac{k_2}{2^2}} |k_2\rangle \otimes \cdots \otimes \sum_{k_n=0}^1 e^{i2\pi j \frac{k_n}{2^n}} |k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{i2\pi \frac{j}{2^1}} |1\rangle \right) \otimes \left(|0\rangle + e^{i2\pi \frac{j}{2^2}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{i2\pi \frac{j}{2^n}} |1\rangle \right) \end{aligned} \quad (6.104)$$

続いて、 k の式(6.101)の2進数表示を用いると

$$\begin{aligned} \frac{j}{2^1} &= j_n \cdot 2^{-1} + \cdots + j_1 \cdot 2^{n-2} = j_n \cdot 2^{-1} + \text{integer} \\ e^{i2\pi \frac{j}{2^1}} &= e^{i2\pi(j_n \cdot 2^{-1} + \text{integer})} = e^{i2\pi j_n \cdot 2^{-1}} = e^{i2\pi 0 \cdot j_n} \\ \frac{j}{2^2} &= j_n \cdot 2^{-2} + j_{n-1} \cdot 2^{-1} + \cdots + j_1 \cdot 2^{n-3} = j_n \cdot 2^{-1} + j_{n-1} \cdot 2^{-2} + \text{integer} \\ e^{i2\pi \frac{j}{2^2}} &= e^{i2\pi(j_n \cdot 2^{-2} + j_{n-1} \cdot 2^{-1} + \text{integer})} = e^{i2\pi(j_n \cdot 2^{-2} + j_{n-1} \cdot 2^{-1})} = e^{i2\pi 0 \cdot j_{n-1} j_n} \\ &\vdots \end{aligned} \quad (6.105)$$

となる。ここで、式(6.103)に従って $j_n \cdot 2^{-1} = 0.j_n$ 、 $j_n \cdot 2^{-1} + j_{n-1} \cdot 2^{-2} = 0.j_n j_{n-1}, \cdots$ としている。これを用いると

$$\begin{aligned} |j\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi jk/2^n} |k\rangle = \cdots = \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{i2\pi \frac{j}{2^1}} |1\rangle \right) \otimes \left(|0\rangle + e^{i2\pi \frac{j}{2^2}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{i2\pi \frac{j}{2^n}} |1\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{i2\pi 0 \cdot j_n} |1\rangle \right) \otimes \left(|0\rangle + e^{i2\pi 0 \cdot j_{n-1} j_n} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{i2\pi 0 \cdot j_1 j_2 \cdots j_n} |1\rangle \right) \end{aligned} \quad (6.106)$$

を得る。式(6.106)では、 n 個の量子ビットが重ね合わせの状態になっている。

図6.17に量子離散的フーリエ変換を実現する量子回路を示す。この図は一見、 $|j\rangle$ を $|k\rangle$ に変換する式(6.95)の逆変換のように見えるが、 $|j\rangle$ の $|k\rangle$ による表現である式(6.95)の表現を求める量子回路である。

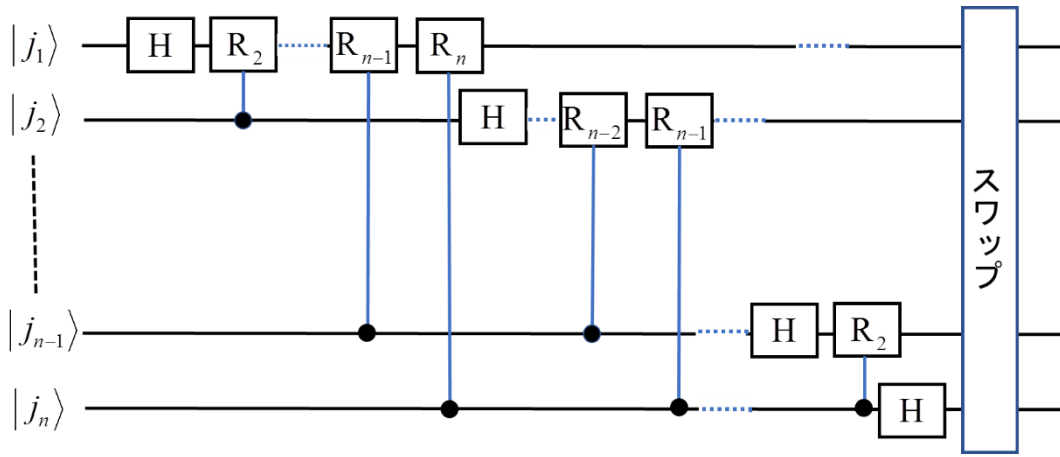


図 6.17 量子離散的フーリエ変換の量子回路. スワップは量子ビットの順番を逆転させるスワップゲートである.

制御回転ゲート R_m は、制御ビットが 0 のときは標的ビットをそのまま通し、制御ビットが 1 のときには標的ビットに以下のような変換を施す：

$$|0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow e^{i\frac{2\pi}{2^m}}|1\rangle \quad (6.107)$$

この量子回路が量子離散的フーリエ変換を実現している.

まず入力は

$$|j_1, j_2, \dots, j_n\rangle = |j_1\rangle \otimes |j_2\rangle \otimes \dots \otimes |j_n\rangle \quad (6.108)$$

第 1 量子ビット $|j_1\rangle$ が最初のアダマールゲート H を通過すると、全体の状態は

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi 0 \cdot j_1}|1\rangle) \otimes |j_2, \dots, j_n\rangle \quad (6.109)$$

となる. このことを確認する. 式 (6.108) は、 $j_1 = 0$ なら明らかに正しい：

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |j_2, \dots, j_n\rangle \quad (6.110)$$

$j_1 = 1$ のとき 2 進法の 0.1 は $0.1_{(2)} = 1 \cdot 2^{-1} = 1/2$ だから、 $e^{i2\pi 0 \cdot j_1} = e^{i2\pi 0 \cdot 1_{(2)}} = e^{i\pi} = -1$ なので

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |j_2, \dots, j_n\rangle \quad (6.111)$$

となるので正しい.

第 1 量子ビットが制御回転ゲート R_2 を通過すると、全体の状態は

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi 0 \cdot j_1 j_2}|1\rangle) \otimes |j_2, \dots, j_n\rangle \quad (6.112)$$

に変換される. これを確認しよう. $j_2 = 0$ のとき R_2 変換は施されないから式 (6.109) がそのまま出力される. したがって

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi 0 \cdot j_1}|1\rangle) \otimes |j_2, \dots, j_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi 0 \cdot j_1 0}|1\rangle) \otimes |j_2, \dots, j_n\rangle \quad (6.113)$$

であるから式 (6.112) は正しい. $j_2 = 1$ のときには、式 (6.107) の変換が第 1 量子ビットに施される. それらは

$$|0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow e^{i\frac{2\pi}{2^m}}|1\rangle \quad (6.107)$$

であるから、全体の状態は

$$\begin{aligned} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1} e^{i\frac{2\pi}{2^2}} |1\rangle \right) \otimes |j_2, \dots, j_n\rangle &= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1} e^{i2\pi 0.01} |1\rangle \right) \otimes |j_2, \dots, j_n\rangle \\ &= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1} |1\rangle \right) \otimes |j_2, \dots, j_n\rangle \end{aligned} \quad (6.114)$$

となり、 $j_2=1$ のときも式(6.112)の変換は正しいことが分かる。

同様に、第1ビットが R_3, \dots, R_n まで通過すると、その出力が

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2, \dots, j_n} |1\rangle \right) \otimes |j_2, \dots, j_n\rangle \quad (6.115)$$

となる。最後にスワップゲートを通過するので、第1量子ビットは第 n ビットとして出力される。式(6.106)と比較すると、第 n 量子ビットの出力となっている。

第2量子ビットについて考える。まずアダマールゲートを通過する。通過後の全体の状態は、第1量子ビットのときと同様に

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2, \dots, j_n} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_2} |1\rangle \right) \otimes |j_3, \dots, j_n\rangle \quad (6.116)$$

となる。さらに、第2量子ビットが R_2, \dots, R_n まで通過すると、全体の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2, \dots, j_n} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_2 j_3, \dots, j_n} |1\rangle \right) \otimes |j_3, \dots, j_n\rangle \quad (6.117)$$

同様な操作を第 n 量子ビットまで続けると、全体の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2, \dots, j_n} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_2 j_3, \dots, j_n} |1\rangle \right) \otimes \dots \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_n} |1\rangle \right) \quad (6.118)$$

スワップゲートを通過すると、式(6.106)の状態が生成されている。すなわち、図6.17の量子回路が量子離散的フーリエ変換を実現している。

図6.17で示した量子離散的フーリエ変換器の計算速度が、式(6.93)のフーリエ変換を古典的に実現する最適アルゴリズム (Fast Fourier Transform; FFT) に比べてどの程度速いかを考える。

FFTの場合、 n ビットで表すことのできる数に必要な論理ゲート数は、 $n2^n$ の程度であることが知られている (古澤明「量子光学と量子情報科学、数理工学社 (2005))。

それに対して、図6.17で示した量子離散的フーリエ変換器では、アダマールゲートが n 個と制御回転ゲートの数が $(n-1)+(n-2)+\dots+1=n(n-1)/2$ で合計 $n+n(n-1)/2=n(n+1)/2$ 個でオーダー n^2 である。したがって、 $n=10$ ビットの場合でも、FFTの方が約100倍ゲート数が多い。もっと大きな数の場合、その差は歴然としている。

量子離散的フーリエ変換は、量子計算機で「あっ」という間に解けるのに通常の古典コンピューターでは時間が掛かり過ぎて事実上解けない場合の例になっている。

以下では具体例を考える。図6.17で示した量子回路を用いて、 $|1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle = |1001\rangle = |9\rangle$ の量子離散的フーリエ変換を求めてみる (図6.18)。

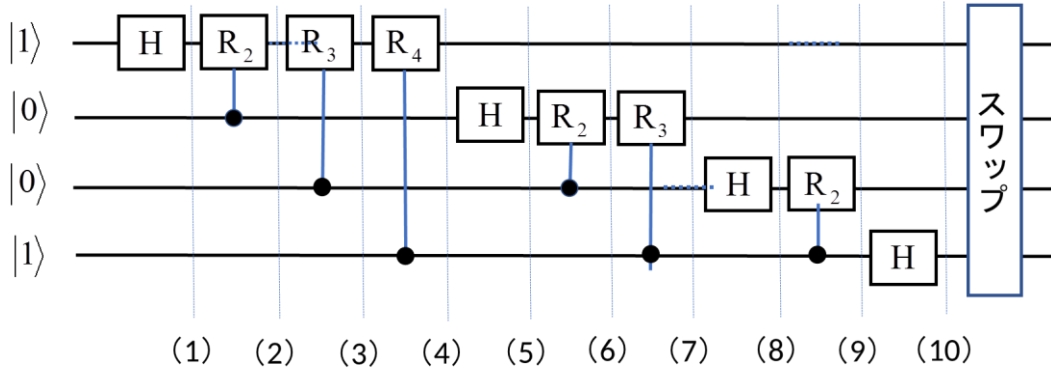


図 6.18 $|1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle = |1001\rangle = |9\rangle$ の量子離散的フーリエ変換を求める．括弧付きの番号は各ステップを示している．

第 1 ビットがアダマールゲートを通過後 (1) の全体の状態は

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \quad (6.119)$$

となる．第 1 量子ビットが制御ゲート R_2 を通過後 (2) の全体の状態は，制御ビットが 0 なので，上式の状態が保たれる．(3) の状態も同様である．しかし，(4) の状態は制御ビットが 1 なので， R_4 ゲートの操作が施される．その結果，(4) の全体の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \quad (6.120)$$

第 2 量子ビットがアダマールゲートを通出した (5) の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |0\rangle \otimes |1\rangle \quad (6.121)$$

である． R_2 ゲートを通過後 (6) の全体の状態は，制御ビットが 0 なので，(5) の状態と同じである．(7) の状態は制御ビットが 1 なので， R_3 の変換が施される．その結果

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}} |1\rangle \right) \otimes |0\rangle \otimes |1\rangle \quad (6.122)$$

第 3 量子ビットがアダマールゲートを通出した (8) の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |1\rangle \quad (6.123)$$

となる．さらに次の R_2 ゲートを通過するとき，制御ビットである第 4 量子ビットは 1 であるので，(9) の状態は

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{2}} |1\rangle \right) \otimes |1\rangle \quad (6.124)$$

第 4 量子ビットがアダマールゲートを通出した (10) の状態では

$$\frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{2}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (6.125)$$

最後にスワップゲートを通過すると

$$\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{2}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{\pi}{4}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle - e^{i\frac{\pi}{8}} |1\rangle \right) \quad (6.126)$$

この式は 4 桁の 2 進数で表される $|k\rangle$ の基底 $|0000\rangle, |0001\rangle, \dots, |1111\rangle$ の $2^4 = 16$ 個に適当な係数を掛けたものの和である．上式を書き直すと

$$\begin{aligned}
|1001\rangle_j &= |9\rangle_j = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{2}}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - e^{i\frac{\pi}{8}}|1\rangle) \\
&= \frac{1}{4} \left(|0000\rangle - e^{i\frac{\pi}{8}}|0001\rangle + e^{i\frac{\pi}{4}}|0010\rangle - e^{i(\frac{\pi}{4} + \frac{\pi}{8})}|0011\rangle \right. \\
&\quad + e^{i\frac{\pi}{2}}|0100\rangle - e^{i(\frac{\pi}{2} + \frac{\pi}{8})}|0101\rangle + e^{i(\frac{\pi}{2} + \frac{\pi}{4})}|0110\rangle - e^{i(\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{8})}|0111\rangle \\
&\quad - |1000\rangle + e^{i\frac{\pi}{8}}|1001\rangle - e^{i\frac{\pi}{4}}|1010\rangle + e^{i(\frac{\pi}{4} + \frac{\pi}{8})}|1011\rangle \\
&\quad \left. - e^{i\frac{\pi}{2}}|1100\rangle + e^{i(\frac{\pi}{2} + \frac{\pi}{8})}|1101\rangle - e^{i(\frac{\pi}{2} + \frac{\pi}{4})}|1110\rangle + e^{i(\frac{\pi}{2} + \frac{\pi}{4} + \frac{\pi}{8})}|1111\rangle \right) \\
&= \frac{1}{4} \left(|0\rangle - e^{i\frac{\pi}{8}}|1\rangle + e^{i\frac{\pi}{4}}|2\rangle - e^{i\frac{3\pi}{8}}|3\rangle + e^{i\frac{\pi}{2}}|4\rangle - e^{i\frac{5\pi}{8}}|5\rangle + e^{i\frac{3\pi}{4}}|6\rangle - e^{i\frac{7\pi}{8}}|7\rangle \right. \\
&\quad \left. - |8\rangle + e^{i\frac{\pi}{8}}|9\rangle - e^{i\frac{\pi}{4}}|10\rangle + e^{i\frac{3\pi}{8}}|11\rangle - e^{i\frac{\pi}{2}}|12\rangle + e^{i\frac{5\pi}{8}}|13\rangle - e^{i\frac{3\pi}{4}}|14\rangle + e^{i\frac{7\pi}{8}}|15\rangle \right)
\end{aligned} \tag{6.127}$$

と書ける。添え字の付いていない $|\square\rangle$ は $|\square\rangle_k$ を意味する。

確かに、式(6.95)：

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi jk/N} |k\rangle \tag{6.95}$$

において、 $|j\rangle = |9\rangle$ とした場合になっている。この式において、例えば $|j\rangle = |9\rangle$ 、 $|k\rangle = |14\rangle$ のとき

$$(e^{i2\pi jk/16})_{j=9, k=14} = e^{i2\pi 9 \times 14/16} = e^{i\pi 252/16} = e^{i\pi(15.75)} = e^{i15\pi} e^{i\frac{3\pi}{4}} = -e^{i\frac{3\pi}{4}} \tag{6.128}$$

となり、 $|j\rangle = |9\rangle$ 、 $|k\rangle = |15\rangle$ のとき

$$(e^{i2\pi jk/16})_{j=9, k=15} = e^{i2\pi 9 \times 15/16} = e^{i\pi 270/16} = e^{i\pi(16.875)} = e^{i16\pi} e^{i\frac{7\pi}{8}} = e^{i\frac{7\pi}{8}} \tag{6.129}$$

となるので、式(6.127)と式(6.95)が等しいことが分かる。

6.5.4 位相推定問題

量子離散的フーリエ変換の応用として、位相推定問題を考える。ユニタリー行列の固有値を量子コンピューターによって計算するアルゴリズムである。ユニタリー行列は、その固有値を必ず複素平面の単位円上に持つ。すなわち、位相推定問題とは以下のような問題である：

「ユニタリー演算子（行列） U が与えられたものとして

$$U|u\rangle = e^{i2\pi\phi}|u\rangle \tag{6.130}$$

という $e^{i2\pi\phi}$ を固有値とし $|u\rangle$ を固有ベクトルとする固有値問題で定義される位相 ϕ を推定する。ただし、固有ベクトルは既知とする」

ここで、 ϕ は $0 \leq \phi \leq 1$ とし、2進数表示では $\phi = 0.\phi_1\phi_2\cdots\phi_n$ であるとする。転置共役行列 U^\dagger がそれ自身の逆であるようなものをユニタリー行列という。

図6.19に位相推定問題を解く量子回路を示す。図の中で、一つ目のグループは第1量子ビットから第 n 量子ビットまでの n 個の量子ビットで、最終的には答えとなる ϕ を出力するので、「解答レジスター」と呼ぶ。

もう一つのグループは、問題の一部である u を表現するので、「問題レジスター」と呼ぶ。未知であるはずの u が使用されているが、このことについては、次節で説明する。 U^k は k 個の U の積を表す。

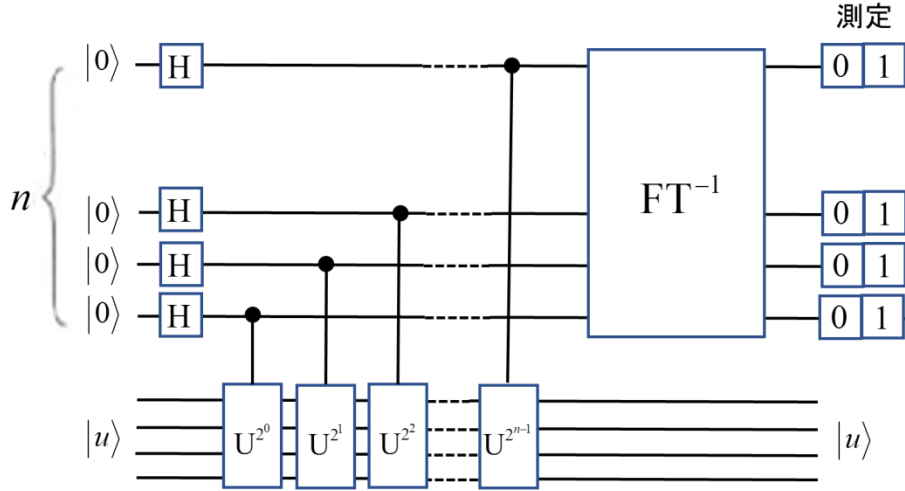


図 6.19 位相推定問題を解く量子回路. FT^{-1} は逆量子離散的フーリエ変換, 測定は 0 か 1 であるかを明らかにする測定である.

解答レジスタの量子ビットがアダマールゲートを通直後の全体の状態は

$$\frac{1}{\sqrt{2^n}}(|0\rangle+|1\rangle)\otimes(|0\rangle+|1\rangle)\otimes\cdots\otimes(|0\rangle+|1\rangle)\otimes|u\rangle \quad (6.131)$$

となる. ここで重要なことを一つ指摘したい. すでに第 6.2.3 節の(2)で説明したように, アダマールゲートを通すことにより, 初期状態 $|0\rangle$ という「確定状態」が $(1/\sqrt{2})(|0\rangle+|1\rangle)$ という「重ね合わせ状態」になることである. このことにより, 量子計算特有の超並列状態になる.

レジスタの一番下にある第 1 ビット $(1/\sqrt{2})(|0\rangle+|1\rangle)$ と問題レジスタのペア $|u\rangle$ の制御 U^{2^0} ゲート通過前の状態 $(1/\sqrt{2})(|0\rangle+|1\rangle)\otimes|u\rangle$ が, ゲート通過で

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes|u\rangle &= \frac{1}{\sqrt{2}}(|0\rangle\otimes|u\rangle+|1\rangle\otimes|u\rangle) \\ &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle\otimes|u\rangle+|1\rangle\otimes U^{2^0}|u\rangle) = \frac{1}{\sqrt{2}}(|0\rangle\otimes|u\rangle+|1\rangle e^{i2\pi\phi}\otimes|u\rangle) = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle e^{i2\pi\phi})\otimes|u\rangle \end{aligned} \quad (6.132)$$

となる. 不思議なことに, 標的ビットの $|u\rangle$ はゲート通過で変化していないで, 制御ビットに変化が起きている.

第 k ビットの場合には はゲート通過で, ひとまず次式のようになる:

$$|u\rangle \rightarrow U^{2^k}|u\rangle = (e^{i2\pi\phi})^{2^k}|u\rangle = e^{i2\pi 2^k\phi}|u\rangle \quad (6.133)$$

ここで, 式(6.130)を用いている. したがって, 解答レジスタの第 k ビットと問題レジスタの U^{2^k} 通過後の状態をまとめて書くと

$$|0\rangle\otimes|u\rangle+|1\rangle\otimes e^{i2\pi 2^k\phi}|u\rangle = |0\rangle\otimes|u\rangle+e^{i2\pi 2^k\phi}|1\rangle\otimes|u\rangle = (|0\rangle+e^{i2\pi 2^k\phi}|1\rangle)\otimes|u\rangle \quad (6.134)$$

となり, 解答レジスタの第 k レジスタが $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ から $\frac{1}{\sqrt{2}}(|0\rangle+e^{i2\pi 2^k\phi}|1\rangle)$ に変化するという不思議なことが起こった. これは驚くべきことであろう.

故に, 制御ゲートを各量子ビットが通過した後の全体の状態は

$$\frac{1}{\sqrt{2^n}}(|0\rangle+e^{i2\pi 2^{n-1}\phi}|1\rangle)\otimes(|0\rangle+e^{i2\pi 2^{n-2}\phi}|1\rangle)\otimes\cdots\otimes(|0\rangle+e^{i2\pi 2^0\phi}|1\rangle)\otimes|u\rangle \quad (6.135)$$

となる。「制御ビットは変化しない」というルールに反しているが論理的な矛盾はない。

話を位相推定問題に戻そう。図 6. 19 の制御 ゲートをすべての量子ビットが通過した後の全体の状態は式 (6. 135) である。ところで、式 (6. 135) の ϕ は 2 進法表示で $\phi = 0.\phi_1\phi_2\cdots\phi_n$ であり

$$e^{2\pi i 2^{n-1}\phi} = e^{i2\pi 2^{n-1}(0.\phi_1\phi_2\cdots\phi_n)} = e^{i2\pi(\phi_1\phi_2\cdots\phi_{n-1}+0.\phi_n)} = e^{i2\pi 0.\phi_n}, e^{2\pi i 2^{n-2}\phi} = e^{i2\pi 0.\phi_{n-1}\phi_n}, \dots, e^{2\pi i 2^0\phi} = e^{i2\pi 0.\phi_1\phi_2\cdots\phi_n} \quad (6. 136)$$

となるので、式 (6. 135) を書き直すと

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{i2\pi 0.\phi_n}|1\rangle) \otimes (|0\rangle + e^{i2\pi 0.\phi_{n-1}\phi_n}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{i2\pi 0.\phi_1\phi_2\cdots\phi_n}|1\rangle) \otimes |u\rangle \quad (6. 137)$$

となる。上式から明らかなように、このときの解答レジスターの状態は、式 (6. 106) で示した量子離散的フーリエ変換の出力状態になっている ($|\phi_1\phi_2\cdots\phi_n\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$ をフーリエ変換した結果になっている)。したがって、図 6. 19 のように量子レジスターを逆量子離散的フーリエゲートに通すと、その出力は $|\phi_1\phi_2\cdots\phi_n\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$ になる。これを測定し、出た答えを 2^n で割れば、最終的に求めたかった答となる。

この計算のステップ数は、量子離散的フーリエ変換と同じオーダーであるから、古典的アルゴリズムに比べて、ステップ数は極端に小さくなり、古典コンピュータにくらべて、超高速で計算できる。

今後の応用のため、あえて 10 進数表記で、このアルゴリズムを書き直して見る。まず、入力状態 $|0\rangle \otimes |u\rangle$ (ただし、0 は 10 進数) は、アダマールゲートにより、解答レジスターの部分が $|0\rangle$ から $|2^n - 1\rangle$ までの重ね合わせになっている。これは、式 (6. 93) と同じで

$$\begin{aligned} & \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 |k_{n-1}\rangle \otimes |k_{n-2}\rangle \otimes \cdots \otimes |k_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 |k_{n-1}k_{n-2}\cdots k_0\rangle \\ &= \frac{1}{\sqrt{2^n}}(|00\cdots 00\rangle + |00\cdots 01\rangle + \cdots + |11\cdots 11\rangle) = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle + \cdots + |2^n - 1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \end{aligned} \quad (6. 138)$$

次に、全体の状態が制御ゲートを通ると、式 (6. 135) と $\sum_{k=0}^1 (|0\rangle + e^{i2\pi 2^k\phi}|1\rangle) = \sum_{k=0}^1 |k\rangle e^{i2\pi k \times 2^k\phi}$ より状態は

$$\begin{aligned} & \left(\frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 |k_{n-1}\rangle \otimes |k_{n-2}\rangle \otimes \cdots \otimes |k_0\rangle \right) \otimes |u\rangle \\ & \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 |k_{n-1}\rangle \otimes |k_{n-2}\rangle \otimes \cdots \otimes |k_0\rangle \otimes U^{k_{n-1} \times 2^{n-1}} U^{k_{n-2} \times 2^{n-2}} \cdots U^{k_0 \times 2^0} |u\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 |k_{n-1}k_{n-2}\cdots k_0\rangle \otimes U^{k_{n-1} \times 2^{n-1} + k_{n-2} \times 2^{n-2} + \cdots + k_0 \times 2^0} |u\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U^k |u\rangle \end{aligned} \quad (6. 139)$$

となる。ただし、これは単に図 6. 19 の制御 U^{2^k} ゲートの部分を 10 進法表記に書き直しただけである。

式 (6. 139) は以下のように変形できる：

$$\begin{aligned} & \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U^k |u\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes e^{i2\pi k\phi} |u\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi k\tilde{\phi}/2^n} |k\rangle \otimes |u\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi \tilde{\phi} k/2^n} |k\rangle \otimes |u\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^N e^{i2\pi \tilde{\phi} k/N} |k\rangle \otimes |u\rangle \end{aligned} \quad (6. 140)$$

ここで

$$\phi = 0.\phi_1\phi_2\cdots\phi_n, \quad \tilde{\phi} = 2^n\phi = \phi_1\phi_2\cdots\phi_n, \quad N = 2^n \quad (6. 141)$$

が用いられた。故に、2進表記のときと同じように、 $\otimes|u\rangle$ 以外の部分 $\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi\tilde{\phi}k/N} |k\rangle$ は、式(6.95)で定義される $|\tilde{\phi}\rangle$ の量子離散的フーリエ変換になっている。故に、2進表記のときと同じように逆量子フーリエ変換ゲートを通せば、解答レジスターに $|\tilde{\phi}\rangle$ が現われ、 $N=2^n$ で割れば、位相推定問題の答になる。逆量子フーリエ変換の具体例は次節で述べる。

実は式(6.140)は量子コンピューターにおける最も重要なことを含んでいる。それは、式(6.140)の1番目と2番目の等号である。前述したように制御 U^k ゲートの演算 U^k は $|u\rangle$ に作用する。しかし、作用後は演算子はただの数 $e^{i2\pi k\phi}$ になってしまう。こうなれば、2番目の等号のように、前に持ってくるができる。

そうすると、演算 U^k が作用したにも関わらず、 $|u\rangle$ はそのまま何をしなかったように見える。一方、本来変化するはずのない、制御ビット $|k\rangle$ に係数 $e^{i2\pi k\phi}$ が現われる。これらは、量子コンピューターの本質であるエンタングルメントの効果である。

つまり、エンタングルしている片方に操作を施すと、空間的に離れているもう片方に影響が及ぶのである。数学的には、演算子から数への変化、さらには係数は移動できるので、量子エンタングルメントの効果が発現する。

6.5.5 位数計算

位数計算は、次節で述べるように、ショアのアルゴリズムを使って素因数分解するためのキーである。位数計算の量子回路では、位相推定問題を解く量子回路が応用されている。この節では、この位数計算の量子回路を考えてみる。位数については、既に第6.5.2節の式(6.75)で示されている。位数計算の量子アルゴリズムを理解することは簡単ではない。そもそも、位数の概念自体が馴染み難い。第6.5.2節の数値例を通して、整数商の剰余の周期性に十分に馴染むことが必要であろう。また、位数計算が演算子の位相推定という固有値問題に変換され、位相推定の一部を成すフーリエ変換を通して推定された位相から位数が推定されるという複雑な道筋を、しっかり理解しなければならない。適切な数値例が理解の大きな助けになろう。

改めて位数の定義を以下に示す：

「互いに素の自然数 x と M ($x < M$) が与えられたとき

$$x^r \bmod M = 1 \quad (6.142)$$

となる最小の自然数 r を x と M の位数という。」

通常のコンピューターで r を計算するのは難しいとされている。量子計算では

$$U_{x,M} |y\rangle \equiv |xy \bmod M\rangle \quad (6.143)$$

を満たす演算子 $U_{x,M}$ があるとし、さらにこの演算子 $U_{x,M}$ の固有値問題が下記のように与えられるとす

$$U_{x,M} |u_s\rangle = e^{i2\pi \frac{s}{r}} |u_s\rangle, \quad s=0,1,\dots,r-1 \quad (6.144)$$

すなわち、 $s=1$ のとき固有状態 $|u_1\rangle$ の固有値 $e^{i2\pi \frac{1}{r}}$ の位相 ϕ は $\phi=1/r$ なので、位数 r が $r=1/\phi$ で与えられる。故に、位数 r の推定には、前節で述べた位相推定問題を解けばよいことになる。

初めに、式(6.143)で定義される演算子 $U_{x,M}$ の性質について述べる。式(6.143)より

$$U_{x,M}^r |y\rangle = U_{x,M} U_{x,M} \cdots U_{x,M} |y\rangle = |x^r y \bmod M\rangle \quad (6.145)$$

となる。 r が位数であるとするので、 p を整数として

$$x^r = pM + 1 \rightarrow x^r y \bmod M = (pM + 1) y \bmod M = y \bmod M \quad (6.146)$$

であるので、 $y < M$ ならば

$$U_{x,M}^r |y\rangle = |x^r y \bmod M\rangle = |y \bmod M\rangle = y \rightarrow U_{x,M}^r = I \quad (6.147)$$

となる。ここで、 I は恒等演算子である。

数値例として $x = 2$ 、 $M = 10$ とすると

$$U_{2,10} |y\rangle = |2y \bmod 10\rangle \quad (6.148)$$

であるので、 $y < M$ のとき

$$\begin{aligned} U_{2,10}^0 |0\rangle &= |0\rangle \rightarrow U_{2,10}^1 |0\rangle = |0\rangle, \dots \\ U_{2,10}^0 |1\rangle &= |1\rangle \rightarrow U_{2,10}^1 |1\rangle = |2\rangle \rightarrow U_{2,10}^2 |1\rangle = |4\rangle \rightarrow U_{2,10}^3 |1\rangle = |8\rangle \rightarrow U_{2,10}^4 |1\rangle = |6\rangle \rightarrow U_{2,10}^5 |1\rangle = |2\rangle, \dots \\ U_{2,10}^0 |2\rangle &= |2\rangle \rightarrow U_{2,10}^1 |2\rangle = |4\rangle \rightarrow U_{2,10}^2 |2\rangle = |8\rangle \rightarrow U_{2,10}^3 |2\rangle = |6\rangle \rightarrow U_{2,10}^4 |2\rangle = |2\rangle, \dots \\ U_{2,10}^0 |3\rangle &= |3\rangle \rightarrow U_{2,10}^1 |3\rangle = |6\rangle \rightarrow U_{2,10}^2 |3\rangle = |2\rangle \rightarrow U_{2,10}^3 |3\rangle = |4\rangle \rightarrow U_{2,10}^4 |3\rangle = |8\rangle \rightarrow U_{2,10}^5 |3\rangle = |6\rangle, \dots \\ &\vdots \\ U_{2,10}^0 |5\rangle &= |5\rangle \rightarrow U_{2,10}^1 |5\rangle = |0\rangle \rightarrow U_{2,10}^2 |5\rangle = |0\rangle, \dots \\ &\vdots \\ U_{2,10}^0 |8\rangle &= |8\rangle \rightarrow U_{2,10}^1 |8\rangle = |6\rangle \rightarrow U_{2,10}^2 |8\rangle = |2\rangle \rightarrow U_{2,10}^3 |8\rangle = |4\rangle \rightarrow U_{2,10}^4 |8\rangle = |8\rangle, \dots \\ U_{2,10}^0 |9\rangle &= |9\rangle \rightarrow U_{2,10}^1 |9\rangle = |8\rangle \rightarrow U_{2,10}^2 |9\rangle = |6\rangle \rightarrow U_{2,10}^3 |9\rangle = |2\rangle \rightarrow U_{2,10}^4 |9\rangle = |4\rangle \rightarrow U_{2,10}^5 |9\rangle = |8\rangle, \dots \end{aligned} \quad (6.149)$$

となる。この場合には、演算子 $U_{2,10}$ の周期 $r = 4$ が $x^r \bmod M = 2^4 \bmod 10 = 6$ を満たしておらず、 $x = 2$ 、 $M = 10$ のときの位数ではない。初期値が 0、5 も例外的であるが、周期 $r = 4$ と見做せる。

他の数値例として $x = 3$ 、 $M = 10$ とすると

$$U_{3,10} |y\rangle = |3y \bmod 10\rangle \quad (6.150)$$

であるので、 $y < M$ のとき

$$\begin{aligned} U_{3,10}^0 |0\rangle &= |0\rangle \rightarrow U_{3,10}^1 |0\rangle = |0\rangle, \dots \\ U_{3,10}^0 |1\rangle &= |1\rangle \rightarrow U_{3,10}^1 |1\rangle = |3\rangle \rightarrow U_{3,10}^2 |1\rangle = |9\rangle \rightarrow U_{3,10}^3 |1\rangle = |7\rangle \rightarrow U_{3,10}^4 |1\rangle = |1\rangle, \dots \\ U_{3,10}^0 |2\rangle &= |2\rangle \rightarrow U_{3,10}^1 |2\rangle = |6\rangle \rightarrow U_{3,10}^2 |2\rangle = |8\rangle \rightarrow U_{3,10}^3 |2\rangle = |4\rangle \rightarrow U_{3,10}^4 |2\rangle = |2\rangle, \dots \\ U_{3,10}^0 |3\rangle &= |3\rangle \rightarrow U_{3,10}^1 |3\rangle = |9\rangle \rightarrow U_{3,10}^2 |3\rangle = |7\rangle \rightarrow U_{3,10}^3 |3\rangle = |1\rangle \rightarrow U_{3,10}^4 |3\rangle = |3\rangle, \dots \\ &\vdots \\ U_{3,10}^0 |5\rangle &= |5\rangle \rightarrow U_{3,10}^1 |5\rangle = |5\rangle, \dots \\ &\vdots \\ U_{3,10}^0 |8\rangle &= |8\rangle \rightarrow U_{3,10}^1 |8\rangle = |4\rangle \rightarrow U_{3,10}^2 |8\rangle = |2\rangle \rightarrow U_{3,10}^3 |8\rangle = |6\rangle \rightarrow U_{3,10}^4 |8\rangle = |8\rangle, \dots \\ U_{3,10}^0 |9\rangle &= |9\rangle \rightarrow U_{3,10}^1 |9\rangle = |7\rangle \rightarrow U_{3,10}^2 |9\rangle = |1\rangle \rightarrow U_{3,10}^3 |9\rangle = |3\rangle \rightarrow U_{3,10}^4 |9\rangle = |9\rangle, \dots \end{aligned} \quad (6.151)$$

となる。この場合には演算子 $U_{3,10}$ の周期 $r = 4$ が $x^r \bmod M = 3^4 \bmod 10 = 1$ を満たしており、 $x = 3$ 、 $M = 10$ のときの位数であることが分かる。初期値が 0、5 も例外的であるが、周期 $r = 4$ と見做せる。

$U_{x,M}$ の周期が y の値に依存しないことは重要である。すなわち、任意の初期状態に次々に $U_{x,M}$ を作用させてゆくと初期状態と無関係に周期性が見られる。さらに、 x が奇数の時には、その周期が位数になる。このことはユニタリ演算子 $U_{x,M}$ の固有値問題の固有値を $e^{i2\pi\phi}$ とすると

$$U_{x,M}^r |y\rangle = (e^{i2\pi\phi})^r |y\rangle = e^{i2\pi r\phi} |y\rangle = |y\rangle \rightarrow e^{i2\pi r\phi} = 1 \rightarrow r\phi = 1 \rightarrow \phi = \frac{1}{r} \text{ or } r = \frac{1}{\phi} \quad (6.152)$$

である。すなわち、位数決定問題が位相決定問題に変換されることが分かる。

ただし、初期値が 5 の時には

$$\begin{aligned} U_{2,10}^0 |5\rangle &= |1\rangle \rightarrow U_{2,10}^1 |5\rangle = |0\rangle \rightarrow U_{2,10}^2 |5\rangle = |0\rangle, \dots \\ U_{3,10}^0 |5\rangle &= |1\rangle \rightarrow U_{3,10}^1 |5\rangle = |5\rangle \rightarrow U_{3,10}^2 |5\rangle = |5\rangle, \dots \end{aligned} \quad (6.153)$$

であるので、周期を 0 と見做せば $5^0 \bmod 10 = 1$ で位数であるが、意味がない位数と考えられる。5 は 10 の約数である。

$x = 7$ の時には

$$U_{7,10}^0 |1\rangle = |1\rangle \rightarrow U_{7,10}^1 |1\rangle = |7\rangle \rightarrow U_{7,10}^2 |1\rangle = |9\rangle \rightarrow U_{7,10}^3 |1\rangle = |3\rangle \rightarrow U_{7,10}^4 |1\rangle = |1\rangle, \dots \quad (6.154)$$

であるので、周期は 4 で $7^4 \bmod 10 = 2401 \bmod 10 = 1$ なので位数である。

$x = 9$ の時には

$$U_{9,10}^0 |1\rangle = |1\rangle \rightarrow U_{9,10}^1 |1\rangle = |9\rangle \rightarrow U_{9,10}^2 |1\rangle = |1\rangle, \dots \quad (6.155)$$

であるので、周期は 2 で $9^2 \bmod 10 = 81 \bmod 10 = 1$ なので位数である。

$x = 5$ は $M = 10$ の約数であるので除外すると、 $M = 10$ と素な奇数は $x = 3, 7, 9$ であり、いずれも位数になっている。

$M = 15$ の時には、 $M = 15$ と素な奇数は $x = 7, 11, 13$ で、それぞれ

$$U_{7,15}^0 |1\rangle = |1\rangle \rightarrow U_{7,15}^1 |1\rangle = |7\rangle \rightarrow U_{7,15}^2 |1\rangle = |4\rangle \rightarrow U_{7,15}^3 |1\rangle = |13\rangle \rightarrow U_{7,15}^4 |1\rangle = |1\rangle, \dots \quad (6.156)$$

$$U_{11,15}^0 |1\rangle = |1\rangle \rightarrow U_{11,15}^1 |1\rangle = |11\rangle \rightarrow U_{11,15}^2 |1\rangle = |1\rangle, \dots \quad (6.157)$$

$$U_{13,15}^0 |1\rangle = |1\rangle \rightarrow U_{13,15}^1 |1\rangle = |13\rangle \rightarrow U_{13,15}^2 |1\rangle = |4\rangle \rightarrow U_{13,15}^3 |1\rangle = |7\rangle \rightarrow U_{13,15}^4 |1\rangle = |1\rangle, \dots \quad (6.158)$$

であるので、周期は 4, 2, 4 になる。いずれも位数になっている。

次に、上の $|u_s\rangle$ を

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-i2\pi j \frac{s}{r}} |x^j \bmod M\rangle \quad (6.159)$$

としてみよう。この時、式 (6.144) が言えることを示そう。そこで、まず下記を示す：

$$U_{x,M} |x^j \bmod M\rangle = |x^{j+1} \bmod M\rangle \quad (6.160)$$

$x^j \bmod M = q$ (q : 自然数) とすると、 $x^j = pM + q$ (p : 自然数) と書けるので、式 (6.143) に $y = x^j \bmod M$ を代入すると

$$U_{x,M} |x^j \bmod M\rangle = |x(x^j \bmod M) \bmod M\rangle = |xq \bmod M\rangle \quad (6.161)$$

となる、また

$$|x^{j+1} \bmod M\rangle = |x \cdot x^j \bmod M\rangle = |x(pM + q) \bmod M\rangle = |xq \bmod M\rangle \quad (6.162)$$

であるので、この 2 式を比べることにより、式 (6.160) が証明された。

式 (6.159) と式 (6.160) を用いて、 $U_{x,M} |u_s\rangle$ を計算する：

$$\begin{aligned}
U_{x,M} |u_s\rangle &= U_x \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-i2\pi j \frac{s}{r}} |x^j \bmod M\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-i2\pi j \frac{s}{r}} U_{x,M} |x^j \bmod M\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-i2\pi j \frac{s}{r}} |x^{j+1} \bmod M\rangle \\
&= \frac{1}{\sqrt{r}} \left(e^{-i2\pi 0 \frac{s}{r}} |x^1 \bmod M\rangle + e^{-i2\pi 1 \frac{s}{r}} |x^2 \bmod M\rangle + \cdots + e^{-i2\pi (r-2) \frac{s}{r}} |x^{r-1} \bmod M\rangle \right. \\
&\quad \left. + e^{-i2\pi (r-1) \frac{s}{r}} |x^r \bmod M\rangle \right) \\
&= e^{i2\pi \frac{s}{r}} \frac{1}{\sqrt{r}} \left(e^{-i2\pi 1 \frac{s}{r}} |x^1 \bmod M\rangle + e^{-i2\pi 2 \frac{s}{r}} |x^2 \bmod M\rangle + \cdots + e^{-i2\pi (r-1) \frac{s}{r}} |x^{r-1} \bmod M\rangle \right. \\
&\quad \left. + e^{-i2\pi r \frac{s}{r}} |x^0 \bmod M\rangle \right) \\
&= e^{i2\pi \frac{s}{r}} \frac{1}{\sqrt{r}} \left(e^{-i2\pi 0 \frac{s}{r}} |x^0 \bmod M\rangle + e^{-i2\pi 1 \frac{s}{r}} |x^1 \bmod M\rangle + e^{-i2\pi 2 \frac{s}{r}} |x^2 \bmod M\rangle + \cdots \right. \\
&\quad \left. + e^{-i2\pi (r-1) \frac{s}{r}} |x^{r-1} \bmod M\rangle \right) \\
&= e^{i2\pi \frac{s}{r}} |u_s\rangle
\end{aligned} \tag{6.163}$$

ここで

$$|x^r \bmod M\rangle = |1\rangle = |1 \bmod M\rangle = |x^0 \bmod M\rangle, \quad e^{-i2\pi r \frac{s}{r}} = e^{-i2\pi s} = 1 = e^{-i2\pi 0 \frac{s}{r}} \tag{6.164}$$

を用いた。

位相推定問題を解く量子回路(図 6.19)を用いれば、位数計算は簡単にできてしまうように見える。但し、図 6.19 の量子回路を動かすには、 $|u\rangle$ が分かっているなければならない。そのためには位数計算ができていなければならない。これは困ったことである。しかし、次の事実を使えば、この難局を乗り切ることができる。その事実とは、下の式である(証明は後出の式(6.166))：

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle \tag{6.165}$$

$|u_s\rangle$ を用意することはできないが、 $|1\rangle$ ならいつでも用意できるので、図 6.20 に示されるように $|u_s\rangle$ の代わりに $|1\rangle$ を問題レジスターの入力とする。

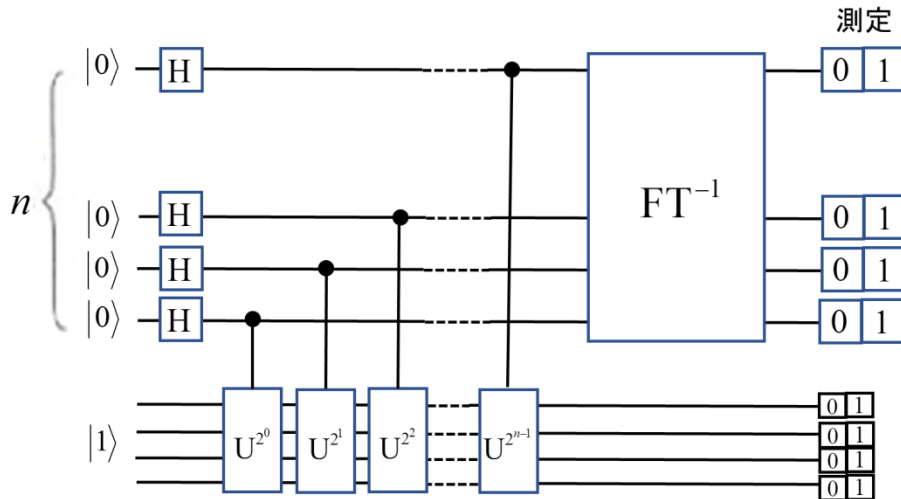


図 6.20 位数を計算する量子回路。 $|u\rangle$ の代わりに $|1\rangle$ を問題レジスターの入力とする。

また問題レジスターの測定も行う。

まず、式(6.165)を確認する。式(6.159)を使うと

$$\begin{aligned}
\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-i2\pi j \frac{s}{r}} |x^j \bmod M\rangle = \frac{1}{r} \sum_{j=0}^{r-1} \left(\sum_{s=0}^{r-1} e^{-i2\pi j \frac{s}{r}} \right) |x^j \bmod M\rangle \\
&= \frac{1}{r} \sum_{j=0}^{r-1} \left(\frac{1 - e^{-i2\pi j}}{1 - e^{-i2\pi \frac{j}{r}}} \right) |x^j \bmod M\rangle = \frac{1}{r} \lim_{j \rightarrow 0} \left(\frac{1 - e^{-i2\pi j}}{1 - e^{-i2\pi \frac{j}{r}}} \right) |x^0 \bmod M\rangle \\
&= \frac{1}{r} r |1 \bmod M\rangle = 1
\end{aligned} \tag{6.166}$$

となる。ここで次式を用いた：

$$\frac{1}{r} \lim_{j \rightarrow 0} \left(\frac{1 - e^{-i2\pi j}}{1 - e^{-i2\pi \frac{j}{r}}} \right) = \frac{1}{r} \lim_{j \rightarrow 0} \left(\frac{i2\pi j}{i2\pi \frac{j}{r}} \right) = 1 \tag{6.167}$$

図 6.20 の量子回路を動かして見よう。これには、10 進法表示の式 (6.140) を使うのが便利である。 $|u\rangle = |1\rangle$ の時に制御 $U_{x,M}^{2^0}, \dots, U_{x,M}^{2^{n-1}}$ ゲートを全体の状態が通ると、全体の状態は式 (6.139) により

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{x,M}^k |1\rangle \tag{6.168}$$

となる。式 (6.165) を使って、これを変形すると

$$\begin{aligned}
\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{x,M}^k |1\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{x,M}^k \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{x,M}^k |u_s\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes e^{i2\pi k \frac{s}{r}} |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi k \frac{s}{r}} |k\rangle \right) \otimes |u_s\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi k \frac{\tilde{s}}{r} / 2^n} |k\rangle \right) \otimes |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi k \frac{\tilde{s}}{r} / N} |k\rangle \right) \otimes |u_s\rangle
\end{aligned} \tag{6.169}$$

この変形では、式 (6.163) および $\tilde{s} = 2^n s$, $2^n = N$ も用いられた。

式 (6.169) より図 6.20 の解答レジスタは、0 から $r-1$ までの r 通りの s を持つ状態の重ね合わせになっていることが分かる。ここで、測定すると、重ね合わせの状態から (測定) の基底状態へ「波束収縮」という量子力学の性質を利用するために、解答レジスターを測定する。この測定により重ね合わせだった解答レジスターは一つの状態になる。また、この波束の収束を用いると、問題レジスターを測定することにより、計算を簡単にすることもできる。

問題レジスターは $|u_s\rangle$ の重ね合わせ状態になっており、その $|u_s\rangle$ 自身も式 (6.169) から $|x^j \bmod M\rangle$ の重ね合わせになっている。このことから、問題レジスタを測定すると、観測された $|x_j \bmod M\rangle$ に波束が収束し、残りの波速は 0 となる。これにより、場合の数が減り、計算が簡単になる (詳しくは次節で述べる)。

このようにしていくと最終的に、解答レジスターの状態は \tilde{s}/r を量子離散的にフーリエ変換したものになるから、最後にこれを逆量子離散的フーリエ変換すると、 \tilde{s}/r が求まり、さらに $N = 2^n$ で割れば s/r が求まる。具体例は次節で説明する。

次の節での見通しを良くするため、もう少し説明を加える。既出の

$$U_{x,M} |y\rangle \equiv |xy \bmod M\rangle \tag{6.143}$$

を仮定すると

$$U_{x,M}^k |1\rangle = |q^k \bmod M\rangle = |x^k \bmod M\rangle \tag{6.170}$$

が成立することが示される。ここで、 $x \bmod M = q$ としている。

まず、 $U_x^k |1\rangle = |q^k \bmod M\rangle$ を数学的帰納法で証明しよう。式 (6.143) で $y=1$ とすると、 $x = pM + q$ であるので

$$y=1 \rightarrow U_{x,M} |1\rangle \equiv |x \bmod M\rangle = |q\rangle \rightarrow U_x^2 |1\rangle = U_{x,M} |q\rangle = |xq \bmod M\rangle = |(pM+q)q \bmod M\rangle = |q^2 \bmod M\rangle \quad (6.171)$$

$$\text{if } U_{x,M}^k |1\rangle = |q^k \bmod M\rangle \rightarrow U^{k+1} |1\rangle = U_{x,M} |q^k \bmod M\rangle = |xq^k \bmod M\rangle = |q^{k+1} \bmod M\rangle \text{ q.e.d.}$$

従って、 $U_{x,M}^k |1\rangle = |q^k \bmod M\rangle$ と言える。次に、 $U_{x,M}^k |1\rangle = |x^k \bmod M\rangle$ については

$$|x^k \bmod M\rangle = |q^k \bmod M\rangle \text{ q.e.d.} \quad (6.172)$$

となる。故に式(6.170)が証明された。

従って、式(6.168)の別の形として

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{x,M}^k |1\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \otimes |x^k \bmod M\rangle \quad (6.173)$$

と言える。

式(6.173)から、制御 $U_{x,M}^{2^k}$ ゲートの行っていることは、実際には x を 0 から $N-1$ までの数でしらみつぶしに、もっと言うともっと重ね合わせの形で x^k を M 割った余りを計算していることになる。制御 $U_{x,M}^{2^0} \dots U_{x,M}^{2^{n-1}}$ ゲート通過後、問題レジスターを測定し、解答レジスターを逆量子離散的フーリエ変換すると、求める位数 r が $x^r \bmod M = 1$ を満たす。

従って、 $x^r = pM + 1$ なので

$$x^{2^r} \bmod M = (pM+1)x^r \bmod M = (pMx^r \bmod M) + (x^r \bmod M) = x^r \bmod M = 1 \quad (6.174)$$

となる。 $x^{nr} \bmod M = 1$ ならば

$$x^{(n+1)r} \bmod M = (p'M+1)x^r \bmod M = x^r \bmod M = 1 \quad (6.175)$$

である。故に、問題レジスターには r の周期で係数の絶対値の 2 乗 (存在確率) が十分にゼロより大きな $|k\rangle$ が現われ、それから最終的に r を求めることになる。

ショアのアルゴリズムでは、素因数分解したい M についての組 (x, r) を見つけ出すことになるが、このときの選択が拙くて、限られた量子ビット数 n 、つまり $|N-1\rangle$ までの中では r の周期が長すぎて、係数の絶対値の 2 乗がどれも十分には大きくなならない場合がある。このような場合は、 x を選び直すことになる。

量子エンタングルメントの効果としては、処理のゲートを極端に減らすことと、波束を収束させ計算する場合の数を減らしていることが挙げられる。

6.5.6 ショアのアルゴリズムの手順

与えられた数を素因数分解するアルゴリズムとは、以下のようなものである。 M が偶数の場合は暗号としての意味がないので除外する。

- (1) 1 から $M-1$ の間で任意に x を選ぶ。もし x と M の最大公約数 $\gcd(x, M)$ が 1 より大きいならば、 $\gcd(x, M)$ を出力する。他の素因数は $M/\gcd(x, M)$ である。
- (2) x, M ($x < M$) の位数 r を計算する ($x^r \bmod M = 1$)。
- (3) もし r が偶数であり、 $x^{r/2} \bmod M \neq 1$ ならば、 $\gcd(x^{r/2}+1, M)$ と $\gcd(x^{r/2}-1, M)$ を計算する。もしこれらのうち一つが M の素因数ならば、それを出力する。だめなら(1)に戻る。

特に、最大公約数を計算するアルゴリズムはユークリッドの互除法として有名である。

従って、ショアのアルゴリズムの要点は、(2) の位数計算を量子コンピューターで行い高速化することである。

具体例を考える。最も簡単な例として、 $M=15$ の場合を量子コンピューターで素因数分解してみよう。 $x=7$ を選ぶことにすると、(1) は当てはまらないので位数計算 (2) へ進む。

$$7^0 \bmod 15 = 1, \quad 7^1 \bmod 15 = 7, \quad 7^2 \bmod 15 = 4, \quad 7^3 \bmod 15 = 13, \quad 7^4 \bmod 15 = 1, \quad \dots \quad (6.176)$$

この例では、 $r=4$ で

$$7^4 \bmod 15 = 1, \quad 7^2 \pm 1 = (50, 48), \quad \gcd(50, 15) = 5, \quad \gcd(48, 15) = 3, \quad 5 \times 3 = 15 \quad (6.177)$$

となり、素因数分解ができた。

同じ $M=15$ の場合に、 $x=3$ あるいは $x=5$ を選んでも良いが、この場合には (1) が適用される。

量子計算の手順の説明に移ろう。まず、解答レジスターに $|0\rangle$ 、問題レジスターに $|1\rangle$ を入力する。アダマールゲートを通過すると、式 (6.138) により全体の状態は

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes |1\rangle \quad (6.178)$$

となる。 $2^n \geq M$ なので、 $M=15$ の場合には $n=4$ とする。制御 U^k ゲートを通過させると式 (6.173) は

$$\begin{aligned} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \otimes U_{7,15}^k |1\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \otimes |x^k \bmod M\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \otimes |7^k \bmod 15\rangle \\ &= \frac{1}{\sqrt{4}} (|0\rangle \otimes |7^0 \bmod 15\rangle + |1\rangle \otimes |7^1 \bmod 15\rangle + |2\rangle \otimes |7^2 \bmod 15\rangle + \dots + |15\rangle \otimes |7^{15} \bmod 15\rangle) \\ &= \frac{1}{\sqrt{4}} (|0\rangle \otimes |1\rangle + |1\rangle \otimes |7\rangle + |2\rangle \otimes |4\rangle + |3\rangle \otimes |13\rangle + |4\rangle \otimes |1\rangle + |5\rangle \otimes |7\rangle + |6\rangle \otimes |4\rangle + |7\rangle \otimes |13\rangle \\ &\quad + |8\rangle \otimes |1\rangle + |9\rangle \otimes |7\rangle + |10\rangle \otimes |4\rangle + |11\rangle \otimes |13\rangle + |12\rangle \otimes |1\rangle + |13\rangle \otimes |7\rangle + |14\rangle \otimes |4\rangle + |15\rangle \otimes |13\rangle) \end{aligned} \quad (6.179)$$

となる。最右辺の部分を見ると、問題レジスター（各項の後ろの要素）が $|1\rangle, |7\rangle, |4\rangle, |13\rangle, |1\rangle, \dots$ であるので、周期が 4 で $r=4$ であることが分かる。従って、 $|k\rangle = |1\rangle, |7\rangle, |4\rangle, |13\rangle$ が出る確率が 0.25 で、これ以外の確率は 0 となる。

次に問題レジスタを測定する。可能性としては、1, 4, 7, 13 を測定結果として得る場合の 4 通りがある。これらは等しい確率で起きるが、**どの場合でも高い確率で最終的な答、つまり素因数分解の結果に行き着くことが知られている**、ここでは、例として、測定結果が 4 だった場合を考えよう。このとき問題レジスターが 4 であるときの解答レジスター（各項の前の要素）は、 $k=2, 6, 10, 14$ であり、解答レジスターの状態は以下になる：

$$\frac{1}{2} (|2\rangle + |6\rangle + |10\rangle + |14\rangle) \quad (6.180)$$

全体に掛かる係数が $1/4$ から $1/2$ に変化したのは、問題レジスターを測定することによって、4 に確定した状態に変化して（波束が収束して）、場合の数が 16 の $1/4$ の 4 に減ったためである。

最後に、解答レジスターを逆量子離散的フーリエ変換する。 $n=4$ の場合の逆量子離散的フーリエ変換の量子回路は図 6.21 のようになる。

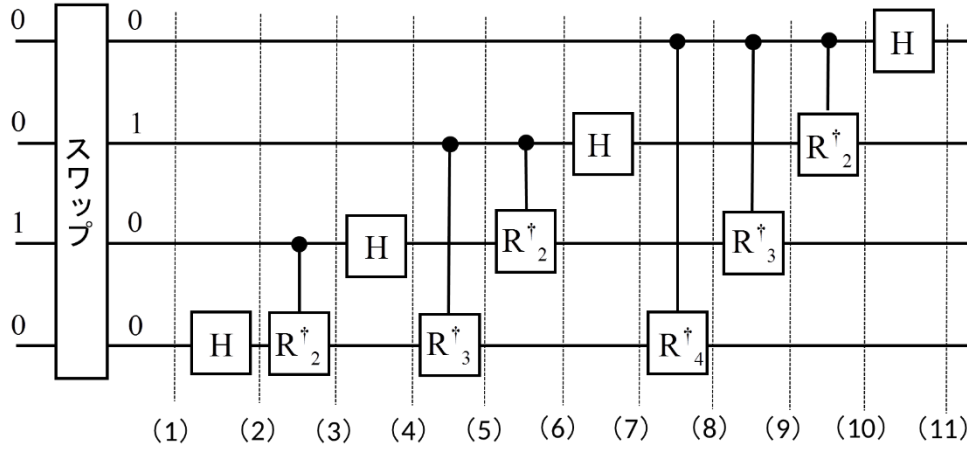


図 6.21 $n=4$ の場合の逆量子離散的フーリエ変換の量子回路. 括弧付きの番号は各ステップを示している.

ここで, 制御 R_m^\dagger ゲートは, 制御 ゲート(式(6.107))の逆変換で, その変換は制御ビットが 1 のときのみに次式の変換が起こり, 0 の場合は素通しである:

$$|0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow e^{-i\frac{2\pi}{2^m}} |1\rangle \quad (6.181)$$

この量子回路に式(6.180)を入力するのであるが, 量子計算機はすべて線形計算なので, $|2\rangle, |6\rangle, |10\rangle, |14\rangle$ を別々に図 6.21 の量子回路に入力し, それらの出力を後で足しても良い. ここでは, $|2\rangle = |0010\rangle$ の場合を実際にやってみる.

まず, 2 は 2 進数で表記すると 0010 であるから, この量子回路への入力は

$$|0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \quad (6.182)$$

である. スワップゲートを通過した状態(1)は

$$|0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle \quad (6.183)$$

次のアダマールゲートを通過した状態(2)は

$$|0\rangle \otimes |1\rangle \otimes |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (6.184)$$

第 3 量子ビットが制御 R_2^\dagger を通過した状態(3)は, 第 3 量子ビットが 0 なので制御ビットである第 4 ビットの状態の如何に関わらずそのままである. 第 3 量子ビットが次のアダマールゲートを通過した状態(4)は

$$|0\rangle \otimes |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (6.185)$$

となる.

第 2 量子ビットが R_3^\dagger ゲートを通過した状態(5)は

$$e^{-i\frac{2\pi}{2^3}} = (1-i)/\sqrt{2} \quad (6.186)$$

であることを用いると

$$|0\rangle \otimes |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + \frac{1-i}{\sqrt{2}}|1\rangle\right) \quad (6.187)$$

となる. 同様に次の R_2^\dagger ゲートを通過した状態(6)は

$$e^{-i\frac{2\pi}{2^2}} = -i \quad (6.188)$$

であるので

$$|0\rangle \otimes |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + \frac{1-i}{\sqrt{2}}|1\rangle\right) \quad (6.189)$$

となる。第2量子ビットがアダマールゲートを通じた状態(7)は、次式のようにになる：

$$|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + \frac{1-i}{\sqrt{2}}|1\rangle\right) \quad (6.190)$$

第1量子ビットは0なので、制御ビットの状態によらずそのままとなり、制御 R_2^\dagger ゲート通過後の状態(10)まではそのままの状態であり続ける。したがって出力される状態は、以下のようにになる：

$$\begin{aligned} |2\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + \frac{1-i}{\sqrt{2}}|1\rangle\right) \\ &= \frac{1}{4}\left(|0000\rangle + \frac{1-i}{\sqrt{2}}|0001\rangle - i|0010\rangle - i\frac{1-i}{\sqrt{2}}|0011\rangle\right. \\ &\quad - |0100\rangle - \frac{1-i}{\sqrt{2}}|0101\rangle + i|0110\rangle + i\frac{1-i}{\sqrt{2}}|0111\rangle \\ &\quad + |1000\rangle + \frac{1-i}{\sqrt{2}}|1001\rangle - i|1010\rangle - i\frac{1-i}{\sqrt{2}}|1011\rangle \\ &\quad \left. - |1100\rangle - \frac{1-i}{\sqrt{2}}|1101\rangle + i|1110\rangle + i\frac{1-i}{\sqrt{2}}|1111\rangle\right) \end{aligned} \quad (6.191)$$

同様に

$$\begin{aligned} |6\rangle &\rightarrow \frac{1}{4}\left(|0000\rangle - i\frac{1-i}{\sqrt{2}}|0001\rangle + i|0010\rangle + \frac{1-i}{\sqrt{2}}|0011\rangle\right. \\ &\quad - |0100\rangle + i\frac{1-i}{\sqrt{2}}|0101\rangle - i|0110\rangle - \frac{1-i}{\sqrt{2}}|0111\rangle \\ &\quad + |1000\rangle - i\frac{1-i}{\sqrt{2}}|1001\rangle + i|1010\rangle + \frac{1-i}{\sqrt{2}}|1011\rangle \\ &\quad \left. - |1100\rangle + i\frac{1-i}{\sqrt{2}}|1101\rangle - i|1110\rangle - \frac{1-i}{\sqrt{2}}|1111\rangle\right) \end{aligned} \quad (6.192)$$

$$\begin{aligned} |10\rangle &\rightarrow \frac{1}{4}\left(|0000\rangle - \frac{1-i}{\sqrt{2}}|0001\rangle - i|0010\rangle + i\frac{1-i}{\sqrt{2}}|0011\rangle\right. \\ &\quad - |0100\rangle + \frac{1-i}{\sqrt{2}}|0101\rangle + i|0110\rangle - i\frac{1-i}{\sqrt{2}}|0111\rangle \\ &\quad + |1000\rangle - \frac{1-i}{\sqrt{2}}|1001\rangle - i|1010\rangle + i\frac{1-i}{\sqrt{2}}|1011\rangle \\ &\quad \left. - |1100\rangle + \frac{1-i}{\sqrt{2}}|1101\rangle + i|1110\rangle - i\frac{1-i}{\sqrt{2}}|1111\rangle\right) \end{aligned} \quad (6.193)$$

$$\begin{aligned} |14\rangle &\rightarrow \frac{1}{4}\left(|0000\rangle + i\frac{1-i}{\sqrt{2}}|0001\rangle + i|0010\rangle - \frac{1-i}{\sqrt{2}}|0011\rangle\right. \\ &\quad - |0100\rangle - i\frac{1-i}{\sqrt{2}}|0101\rangle - i|0110\rangle + \frac{1-i}{\sqrt{2}}|0111\rangle \\ &\quad + |1000\rangle + i\frac{1-i}{\sqrt{2}}|1001\rangle + i|1010\rangle - \frac{1-i}{\sqrt{2}}|1011\rangle \\ &\quad \left. - |1100\rangle - i\frac{1-i}{\sqrt{2}}|1101\rangle - i|1110\rangle + \frac{1-i}{\sqrt{2}}|1111\rangle\right) \end{aligned} \quad (6.194)$$

式(6.191)～(6.194)を加えると $|0000\rangle$ 、 $|0100\rangle$ 、 $|1000\rangle$ 、 $|1100\rangle$ は同符号で足し合うが、他のものは消し合ってしまうので、式(6.180)の状態を逆量子フーリエ変換した結果は

$$\frac{1}{2} \cdot \frac{1}{4}(4|0000\rangle - 4|0100\rangle + 4|1000\rangle - 4|1100\rangle) = \frac{1}{2}(|0\rangle - |4\rangle + |8\rangle - |12\rangle) \quad (6.195)$$

となる.

位数計算問題を解くためには、解答レジスターを測定する. 測定の結果として得られる値は、0, 4, 8, 12 の場合の 4 通りである. 位数を得るには、これらを 2^4 で割る必要があるが、その結果 0, $1/4$, $2/4$, $3/4$ の 4 通りの場合がある. 偶数の位数 r は 4 以下の偶数と考えられるが、 $r = 2$ は $7^2 \bmod 15 = 4 \neq 1$ だから位数でない.

従って、このような位数計算で量子コンピューターを用いて行くと、答として得られる最小の自然数は 4 である. 4 を得た場合 $7^4 \bmod 15 = 1$ であるから位数を得たことになり、最終ステップである (3) に進む.

$r = 4$ は、明らかに (3) の条件を満たすから

$$\gcd(7^{4/2} - 1, 15) = \gcd(48, 15) = 3, \quad \gcd(7^{4/2} + 1, 15) = \gcd(50, 15) = 5 \quad (6.196)$$

を得る. $3 \times 5 = 15$ であるので、15 の素因数分解が得られた.

6.6 あとがき

物理学者ファインマンは、「古典計算機では量子現象のシミュレーションはできない」と述べ、量子計算の必要性を鋭く指摘した. ファインマンが指摘してから半世紀にも満たないが、今日の量子計算の発展は正に驚嘆すべきと言えよう. ただし、まだエラー対策が十分でないので、完成にはほど遠いといえよう.

量子計算は量子力学固有の考え方である「重ね合わせ」と「量子もつれ」に立脚している. 2022 年のノーベル物理学賞の受賞対象は量子もつれで、アラン・アスペ、ジョン・クラウザーそしてアントン・ツァイリンガーに授与された. アインシュタイン等の「量子もつれを許す量子力学は不完全である」とする主張が公式に否定された.

アインシュタインの主張には、彼の強い個人的願望が背景にあると思われる. ニールス・ボーアは「物理学の目的は自然を正確に記述することである」と指摘したが、興味深いことである. 物理学は哲学でもないし、文学でも宗教でもない. 量子力学固有の考え方である「重ね合わせ」や「量子もつれ」を理解しようとしても、真の意味での理解に到達することはできないであろう. 我々の直接的な経験とあまりにもかけ離れているからである. しかし、宗教における神と同様に恩恵にあずかろうとするならば、受け入れるしかない.

量子計算や量子情報処理を通して、人類の文明は大きな飛躍を遂げられると思われるが、それがどのようなものであるかは我々の想像を超えている. 人類に大きな幸せをもたらしてくれることをひたすら願う.

第 6 章の参考文献

- [6-1] 宮野健次郎, 古澤 明, 量子コンピューター入門, 日本評論社 (2008).
- [6-2] 中山 茂, 量子アルゴリズム, 技報堂出版 (2014).
- [6-3] 西森秀稔, 大関真之, 量子アニーリングの基礎, 共立出版 (2020).

第7章 量子計算機実現の方式概要

古典計算機(量子計算機に対して現在主流の計算機は“古典”と呼ばれている)はある時点で0と1のどちらかの値をとる“ビット”の連なったデータで構成されている。これに対し量子計算機では同時に“0”でもあり“1”でもある状態をとる“量子ビット”より構成されている。1量子ビットに着目すれば、“0”であると同時に“1”であるという“重ね合わせ状態”が実現される。そして量子ビット数の増加に伴い2の n 乗個の状態を同時に表現可能となる。また2量子ビットに着目すれば、一方の量子ビットの状態を測定すれば他方の量子ビットの状態も同時に決まるという“量子もつれ”が実現される。量子計算機ではこの量子重ね合わせと量子もつれといった量子力学的な現象を組合せて並列的に広がった無数の量子状態を同時に操作することで高速計算を可能にしている。古典計算機が古典力学に基づいた計算機であるのに対し、量子計算機は量子力学に基づいた技術により実現された計算機である。

2000年に量子計算機を実現するために物理学者デイヴィッド・ディヴィンチェンゾは量子ビットが満たすべき以下の5つの条件(ディヴィンチェンゾのクライテリアと呼ばれる)を提示した。これらの条件は量子計算機の動作原理や実装に関する基本的な指針となっている。

- (1) 多量子ビットへの拡張性(量子ビットのビット数を増やせる)
- (2) 状態の初期化(量子ビットの初期化が可能)
- (3) 十分に長いコヒーレンス時間(量子重ね合わせ状態を計算に必要な時間維持可能)
- (4) 任意の量子ゲートの実現(量子ゲートの必要十分な種類を実現可能)
- (5) 演算結果の読み出し(人間による測定が可能)

7.1 量子計算機 5 方式概要

表 1.1 量子計算機 5 方式概要

項目	超電導	光	イオントラップ	シリコン	冷却原子
会社	Google, IBM 中国 科学技術大学	東京大学+NTT 米国サイクオンタム 中国 科学技術大学	米国 IONQ(イオンキュー)社 米国 Honeywell(ハネウェル)社	米国 Intel 米国 IBM 産総研, 理研	米国 Atom Computing 仏国 Pasqal
方式概要	超伝導材料を用いた電子回路上でジョセフソン接合というトンネル接合素子を用いて量子ビットを実現する方式	電磁波の一種で量子力学の法則に従う光の粒子（光子）を使って量子情報を表現し操作する方式	物質を構成する原子や分子をイオン化させた状態(イオン化粒子)にして電場や磁場中で制止させ, このイオン化粒子を計算単位のビットとして計算に応用する方式	量子ビットとして使う電子をシリコンチップ上の量子ドットに閉じ込めることで電子のスピンを制御する方式	光ピンセットを用いてミクロン間隔で整列させた冷却原子をベースにレーザー光で原子を冷却・捕捉する方式
長短概要	超伝導体は電気抵抗がゼロになる物質でクーパー対という電子のペアが量子ビットとして機能する。超伝導回路はマイクロ波を用いて高速で高精度な操作が可能であるが低温環境が必要で冷却コストが高いという欠点がある。	光子は電磁場の量子で偏光や位相などの性質を利用して情報を表現する。特殊な環境では光方式は量子ビット間の相互作用が強く大規模化が可能であるが, 量子ビットの操作や測定が困難で高精度な光源や検出器が必要である。	イオンは荷電粒子で電磁場を用いて自由空間内に閉じ込めて保持する。イオントラップ方式はレーザーを用いて長時間のコヒーレンスと高い操作精度が実現可能であるが, 量子ビット数の増加に伴うトラップ電圧やレーザー制御の複雑化が課題である。	既存の半導体製造技術に応用可能であり多数の素子をチップに集積化可能であるが, 量子ビットの制御が難しいことや温度やノイズに敏感である。	容易に大規模化が可能な点や高コヒーレンスな点(量子の波としての純度が高い点)において画期的な潜在能力があるが, 極低温環境が必要なことやレーザー光の安定性が課題である。

7.2 超電導方式

超伝導材料を用いた電子回路上でジョセフソン接合というトンネル接合素子を用いて量子ビットを実現する方式である。超伝導量子ビットは、外部から印加した電圧や磁場によってジョセフソン効果(弱く結合した2つの超伝導体の間の超伝導電子対のトンネル効果)によって流れる超伝導電流を制御する。この量子ビット(トランズモン)は非線形インダクタであるジョセフソン接合とキャパシタの単純な並列共振回路で構成される。この量子状態はマイクロ波パルスを用いて操作可能である。

ジョセフソン効果は、弱く結合した2つの超伝導体の間に、超伝導電子対のトンネル効果によって超伝導電流が流れる現象である。図 7.2.1 に示すジョセフソン接合(二つの超伝導体の間に絶縁体などの障壁がある接合)において、障壁層が極めて薄い時に超伝導体間に超伝導電流(ジョセフソン電流)が流れる現象である。超伝導状態の物質はその内部で、全てのクーパー対がボースアインシュタイン凝縮により全体として1つの巨大な電子対として振る舞う。つまりこのとき超伝導を示す電子の物質波の位相は巨大量子化によって物質の隅々まで全く同じ状態になる[7-1]。



図 7.2.1 ジョセフソン効果

図 7.2.2 に示すように、超電導方式の量子ビットでは極低温まで冷却したこのジョセフソン接合を用いた微小リングにおいて発生する右回りと左回りの電流が共存する量子状態が実現される。すなわち右回りの”1”と左回りの”0”が重ね合わさった量子ビットを表現できる。この電流は超伝導ループに印可された磁束によって変化しその結果エネルギー準位が変化する。これは磁束量子ビットと呼ばれ 2002 年にオランダのデルフト工科大学で技術開発された。この磁束量子ビットは環境ノイズの影響を受けにくくコヒーレンス時間が長くなった。さらに 2007 年に米イエール大学で静電容量を増加させることにより電荷ノイズを軽減したトランズモンという技術が開発された。現在の超電導型の主流は電荷ノイズ耐性が高くコヒーレンス時間が長いこのトランズモン型の磁束量子ビットである。量子もつれは、2つの量子ビットの状態が相互依存することによって実現される[7-2]。

量子ビット操作は、インダクタが非線形なトランズモンの離散準位のうちの最低二準位(左右回りの電流の重なりの違いに対応)を量子ビットとして使用し、この準位間隔に共鳴するマイクロ波パルスを照射することにより実現される。また量子ビット読み出しは、量子ビットと分散的に結合した共振器の共振周波数が量子ビットの状態に依存することを利用して実現される。非常に低雑音なジョセフソンパラメトリック増幅器が必要となる。

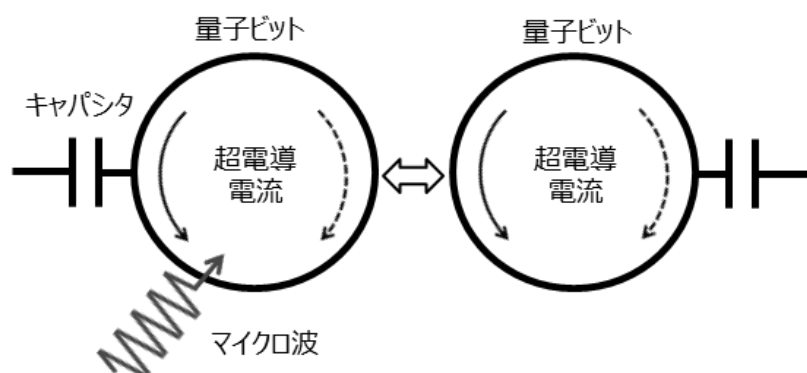


図 7.2.2 超電導量子ビット

<長所>

- ・ マイクロ波 (周波数が数 GHz の電磁波) は超伝導回路の共振周波数と一致するため、マイクロ波照射により量子重ね合わせやもつれ現象を発生させ高速な量子演算を実現可能
- ・ 超伝導回路は低温環境動作のため外部ノイズや熱の影響を受けにくく量子コヒーレンス (量子ビットが重ね合わせ状態を保つ時間) が長いので高精度な量子演算が可能
- ・ 超伝導回路は半導体技術と互換性があり既存の製造設備や工程を利用して小型化し多数の量子ビット集積が可能

<短所>

- ・ 絶対零度に近い低温環境が必要。そのために希釈冷凍機という特殊な装置を使用し量子ビットを 10 ミリケルビン程度に冷却。希釈冷凍機は高価で消費電力が大きいため冷却コストは非常に高価
- ・ 量子ビット数の増加に伴って冷却範囲が広がるため冷却コストはさらに高価

7.3 光量子方式

電磁波の一種で量子力学の法則に従う光の粒子 (光量子) を使って量子情報を表現し操作する方式である。光の異なる偏光状態 (振動方向) を共存させることにより量子重ね合わせを実現している。例えば、45 度偏光は水平偏光 “0” と垂直偏光 “1” の等しい重ね合わせである。2 つ以上の光量子ビットの光の偏光状態が相互依存することによって量子もつれが実現される。

(1) 従来の典型的な光量子計算機方式

従来方式の計算回路 (光パルス 1 入力 1 出力) では、まず情報を持つ光パルス 1 個を別の補助的な光パルスと部分透過ミラーで混ぜ合わせ 2 つの光パルスを量子もつれの状態にする。次にその量子もつれの片方を測定し測定値を得る。最後にその測定値の値に応じてもう片方の光パルスに操作を行い計算結果を得る。図 7.3.1 に示すように、補助光パルスの種類、部分透過ミラーの透過率 (ブロッホ球上の y 軸周りの回転)、位相シフタのシフト量 (z 軸周りの回転) などを変えることで計算の種類を変える。この計算回路は量子テレポーテーションの原理に基づいている。大規模な計算では光回路が多数必要であった [7-3]。

(2) 1 ループ方式

NTT と東京大学が光量子プロセッサを開発。光回路にループ構造を持たせて、量子テレポーテーション回路の無限回繰り返して大規模な量子演算を実行可能とした。①2 光パルス間の量子もつれ合成、②片方の光パルスの測定、③もう片方の光パルスへの操作の一連の手順で四則計算などの単純な計算を 1

回実行する。そして回路構成要素をナノ秒精度で時間同期させて切替える。切替パターン変更により計算の種類や繰り返し回数を制御する。大規模な計算でも最小規模の光回路で実現可能となる[7-3]。

この1ループ方式ではミラーの透過率や光スイッチのオン/オフ等を時々刻々と切替えることで計算を行う汎用性と、一連の手順を繰り返すことにより何ステップも計算を継続できる拡張性を持つ。量子もつれを生成するための補助光パルスとしてスクイズド光(ある位相での揺らぎが真空場よりも小さくなった量子力学的な効果を持つ光)を使用する[7-3]。

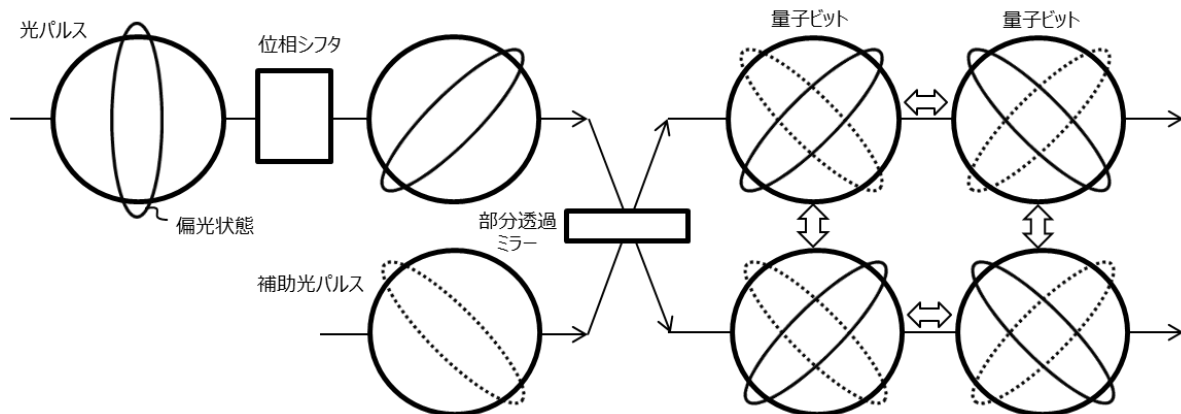


図 7.3.1 光量子ビット計算回路

〈長所〉

- ・ 常温・大気中で動作が可能
- ・ 光を使った量子通信との相性の良さ
- ・ 高クロックによる高速な計算処理が可能
- ・ 特殊な環境では量子ビット間の相互作用が強く大規模化が容易
- ・ 光の“連続量”により 0, 1 の 2 値以上を表現可能

〈短所〉

- ・ 光子は相互作用が弱く、他の物質とほとんど影響し合わないため、光子同士のエンタングルメントや論理演算を実現するには高度な技術が必要
- ・ 光子の測定には検出器が必要であるが検出器は高価で大きく消費電力も大きいいため大規模化に課題

7.4 イオントラップ方式

物質を構成する原子や分子をイオン化させた状態(イオン化粒子)にして電場や磁場中で制止させ、このイオン化粒子を計算単位のビットとして計算に応用する方式である。各イオンはエネルギーの低い状態(“0”：基底状態)とエネルギーの高い状態(“1”：励起状態)の2状態に分かれイオントラップ量子ビットを実現可能である。これらイオントラップ量子ビットに側面からレーザー光線を照射することにより、基底状態と励起状態の間を移行させたり、両者の量子重ね合わせ状態を作り出すことが可能である。2つのイオンを捕獲しイオン間のクーロン相互作用を使用してイオンの量子状態を相互依存させ量子もつれを形成することが可能である。

イオン化粒子を空間中に制止(捕獲)する方法として以下が報告されている[7-4]。

(1) ペニングトラップ(Penning trap)方式

静磁場と静電場によってイオンを捕獲する方式である。真空にした装置内にイオン化させやすい希ガ

スなどの元素を注入しイオン化させ、外部電極からの電場と磁場によりイオン化粒子の振る舞いを限定させ捕獲する方法である。強力な電場と磁場をかけることが必要なために装置が大型となる。

(2) パウルトラップ(Paul trap)方式 (高周波イオントラップ)

静電場と高速で周期的に変動する高周波(RF)電場により x - y 平面の電場をゼロとする個所を z 軸方向に作ることでイオン化粒子を捕獲する方法である。線形四重極ロッド(電極)に電圧を加えることで空間ポテンシャルが生成され、中心の最もポテンシャルが低い場所にイオン化粒子を捕獲。イオン量子ビットの初期化や制御は光(レーザ)との相互作用を用いて実現する。光との相互作用を引起すため光の波長以下の小さな領域にイオンが閉じ込められる。イオン化粒子が高周波加熱されるため安定性に欠けるが装置を小型化可能となる。

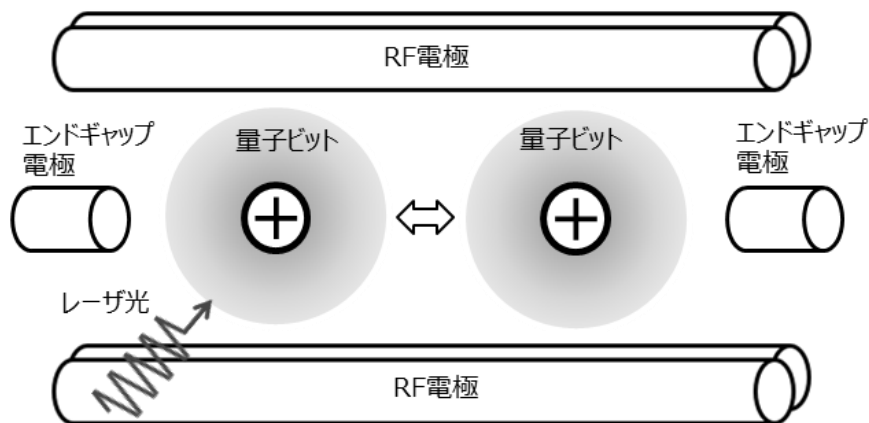


図 7.4.1 イオントラップ量子ビット

<長所>

- ・ ディヴィンチェンゾのクライテリアをクリア
- ・ 常温で稼働させることが可能
- ・ 浮遊するイオンを使用するため構造物からのノイズの影響が少ない
- ・ 構造的に比較的小型化が可能

<短所>

- ・ トラップ電圧(イオンを横一列に並べるために必要な電圧)は並べるイオン数増でトラップ電圧も増大しトラップの安定性が低下。そのため一つのトラップに数百～数千のイオンを並べて制御するのは非常に難しく現在の技術では実現不可
- ・ レーザー制御(イオン状態操作のために必要なレーザーの周波数や強度を調整)は量子ビット数増でレーザー制御も複雑化し誤り率が上昇。そのため高精度なレーザー制御システムや誤り訂正技術が必要

7.5 シリコン方式

量子ビットとして使う電子をシリコンチップ上の量子ドットに閉じ込めることで電子のスピンを制御する方式である。図 7.5.1 に示すように、シリコン量子ビットはシリコン上に 10nm 程度の領域(量子ドット)を形成し閉じ込めた電子で磁場により電子スピン(電子の持つ微小な磁気モーメントで上向きと下向きの二つの状態を取る)の 2 準位系をつくり量子重ね合わせを実現している。量子ドット間のトンネル障壁の大きさに依存するスピンの相互作用、すなわち交換結合の電氣的制御によって量子もつれを実現している。マイクロ波を使って電子状態を操作(重ね合わ)して演算することが可能である。

産総研は、2021年に半導体製造技術を応用してFinFET構造を採用したシリコン量子ビットを日本で初めて作製している。また理研は2022年に3電子による量子誤り訂正技術を開発している。また表7.5.1に示すように、シリコン量子ビットは最先端の半導体プロセスを使用可能であり、超電導やイオントラップ方式と比較して集積度や動作温度に優れている[7-5]。

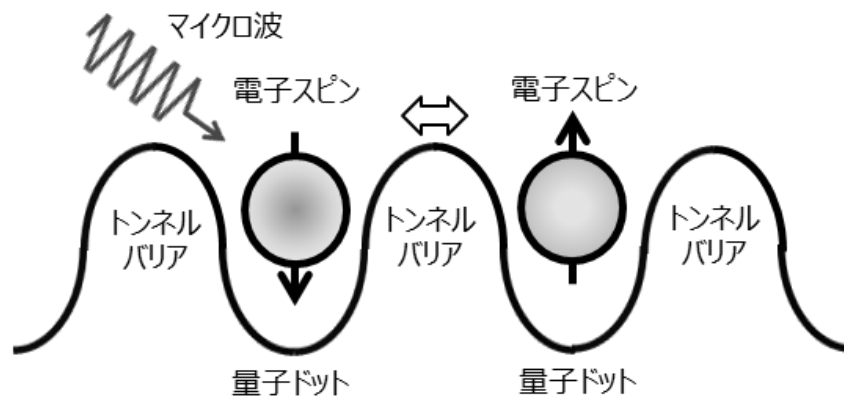


図 7.5.1 シリコン量子ビット

表 7.5.1 シリコン方式と主要方式の性能比較 [7-5]

	シリコン	超電導	イオントラップ
量子ビットサイズ	100 [nm]	10 [um]	1-10 [um]
最大集積度	100 万ビット	数千ビット	数千ビット
動作温度	10K 以下	10mK 以下	10-300K
集積ビット数	3	127	20
会社	理研	米 IBM	米 Quantinuum

＜長所＞

- ・シリコンチップ上の半導体微細化技術により量子ドットのサイズや位置を微細に制御することで多数の量子ビットを集積可能
- ・シリコンは不純物や欠陥が少なく電子のスピンの外部ノイズに影響されにくいため長いコヒーレンス時間を実現可能

＜短所＞

- ・シリコン上に形成した量子ドットに閉じ込めた電子のスピンの向きをマイクロ波やレーザー光等で制御及び測定するには高度な技術が必要
- ・量子ビットは温度や外部ノイズに敏感であり、シリコン基板を構成する原子が量子ドット内の電子に影響を与えないようにするために、シリコン基板を極低温に冷凍するための冷凍機が必要となり全体的な構造が複雑化

7.6 冷却原子方式

光ピンセット（強く集光された光双極子によるトラップの配列）を用いてミクロン間隔で整列させた冷却原子をベースにレーザー光で原子を冷却・捕捉する方式である。図 7.6.1 に示すように、冷却原子

量子ビットは冷却されたルビジウム原子やリユドベリ原子(最外殻電子が高い主量子数 n [n =数 10～100]に励起された原子)によりエネルギーの 2 準位系を制御し量子重ね合わせを実現している。電気の偏りを持つ 2 つ以上の原子が相互作用することで量子もつれを実現している。

世界最速の 2 量子ビットゲート(基本演算要素)操作が実現されている。極低温に冷却した 2 つの原子を μm レベルに近づけ、特殊なレーザーを当てて操作する。10 ピコ秒(1000 億分の 1)だけ光る超高速のパルスレーザーの使用で、6.5 ナノ秒で動作する世界最速の 2 量子ビットゲート(制御ゲート)が実現されている。これはノイズの時間スケールより 2 桁以上速いためノイズの影響をほぼ無視できるのが特徴である(米 Google の 2020 年実現の超電導方式では 15 ナノ秒のゲート時間) [7-6]。

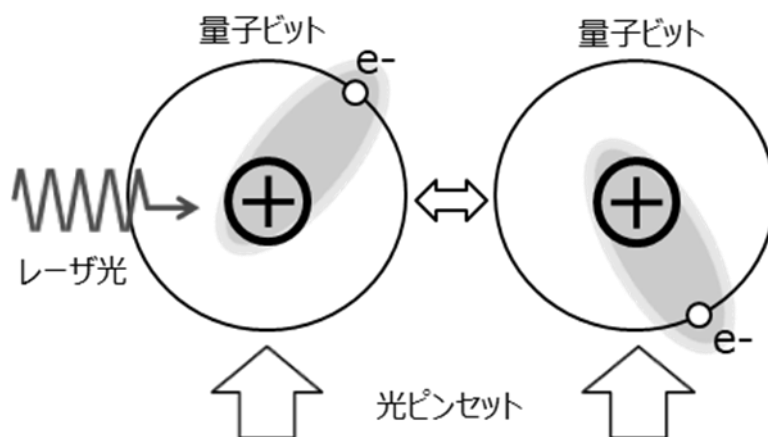


図 7.6.1 冷却原子量子ビット

<長所>

- ・量子ビットの品質が均一
- ・コヒーレンス時間が長い
- ・光ピンセットによって大規模化し易い
- ・量子ビットの全結合の可能性高い

<短所>

- ・原子を極低温に冷却するためには高価で消費電力が大きい冷凍機が必要
- ・レーザー光の周波数や強度が揺らぐと量子ビットの操作や測定に誤差が生じる可能性がありレーザー光の安定性を高めるため高精度な制御装置やフィードバックシステムが必要

7.7 量子計算機の実現方式別の将来展望

1947 年にトランジスタが発明されて以来、トランジスタ技術を応用した古典計算機は飛躍的に発展し続けてきた。1965 年にインテル社のゴードン・ムーアは 1 つの計算機チップに集積できるトランジスタ数が 2 年毎に倍増するムーアの法則を提唱した。その法則に則り古典計算機の演算性能を支えるマイクロプロセッサやメモリ等の性能が今日まで向上し続けている。但しその代償として消費電力増大という弊害が生じていることも確かである。2021 年に米国 IBM が 2nm(ゲート長)の半導体プロセスを開発し 2025 年には量産開始する予定であるが、この微細化では電子等の量子力学的な現象を考慮した制御技術開発が難しくなってくることが指摘されている。これ以上半導体プロセス微細化は困難と言われておりムーアの法則も終焉を迎えつつあると言えるであろう。

これに対し量子計算機は 2010 年頃までは地道に 1, 2 量子ビットの研究が行われてきた。そして 2015

年頃から急速に多量子ビットの量子計算機が開発されている状況である。一つは量子ゲートを利用した汎用的な量子計算機であり、もう一つはイジングモデルによる最適化問題に特化した量子アニーリング方式の量子計算機である。前者は日本でも 2023 年に理研が埼玉県和光市に超電導方式の 64 量子ビットの計算機を稼働させている。後者はカナダの D-Wave 社が先行しており超電導方式の 5,000 量子ビットの計算機を製造している。これら量子計算機においてはまだ誤り訂正技術が確立されておらず、量子計算機の現時点のフェーズは古典計算機における真空管の時代と言われている。現時点ではハードウェア的な故障やエラーが発生し易い状況を前提に量子計算機を使いこなす必要がある。前節までで説明してきた通り超電導方式以外に光方式、イオントラップ方式、シリコン方式、冷却原子方式等様々な方式の量子計算機が研究されている。これら量子ビット実現方式の進化や組合せの発展等により量子計算機においてトランジスタ相当のより安定した量子ビット実現方式の発明が期待される。

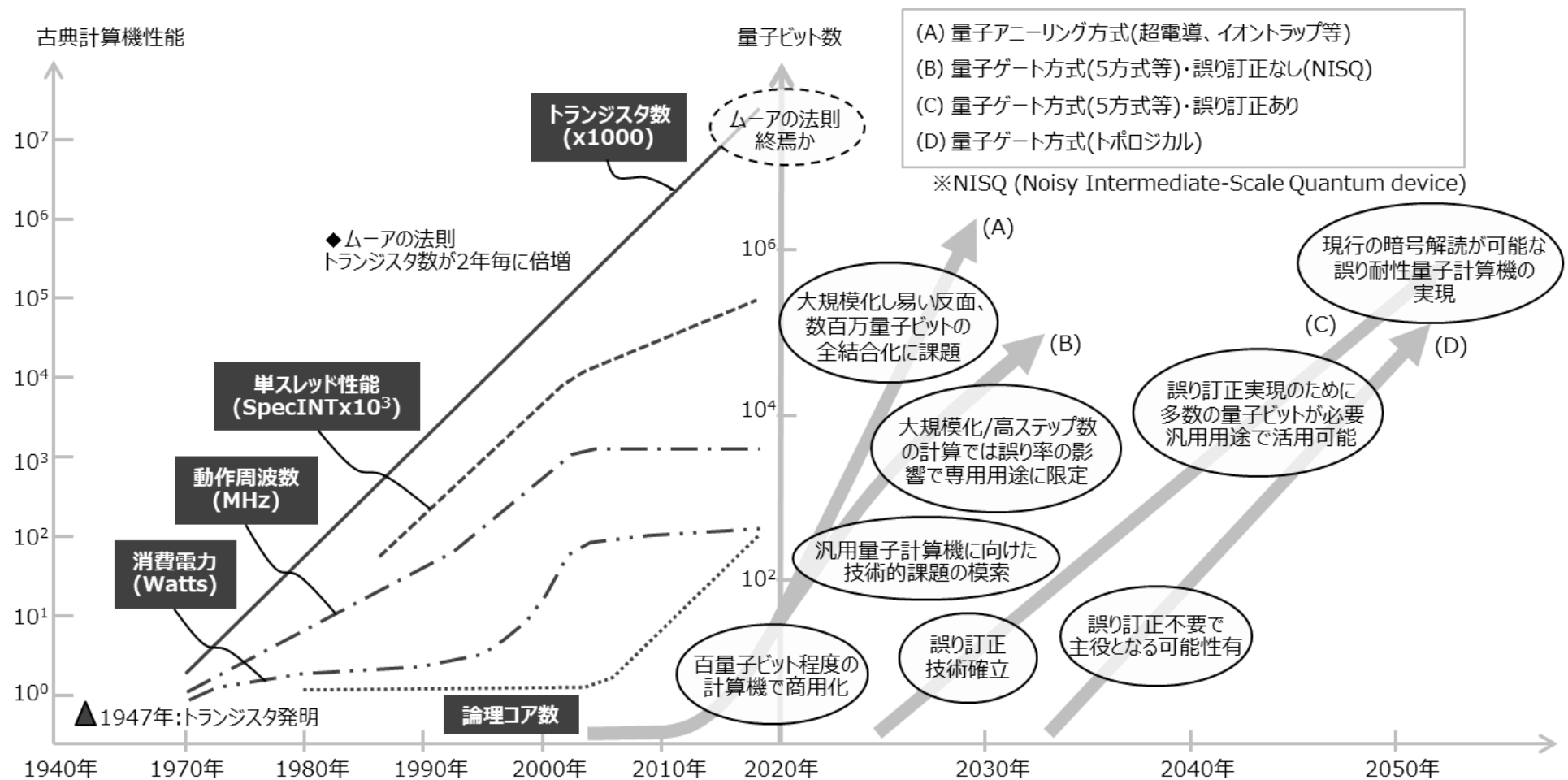


図 7.7.1 量子計算機の実現方式別の将来展望 [7-7], [7-8]

第 7 章の参考文献

[7-1] <https://ja.wikipedia.org/wiki/ジョセフソン効果>, ウィキペディア

[7-2] https://www.aist.go.jp/aist_j/press_release/pr2021/pr20210920/pr20210920.html, 掲載日 : 2021/09/20, シリコン基板を用いた窒化物超伝導量子ビットの開発に成功

[7-3] <https://www.itmedia.co.jp/news/articles/2111/15/news114.html>, 2021/11/15, ITmedia NEWS, 東大、光量子計算の万能プロセッサを開発「究極の大規模光量子コンピュータ」実現へ

[7-4] <https://www.qmedia.jp/basic-of-iontrap/>, 2019/8/28, イオントラップの原理と冷却イオン量子ビット

[7-5] <https://xtech.nikkei.com/atcl/nxt/column/18/01537/00425/>, 2022/8/26, 日経 XTECH, シリコン量子ビット : 汎用量子コンピューター実現へ期待の星

[7-6] <https://xtech.nikkei.com/atcl/nxt/column/18/00001/07208/>, 2022/10/4, 日経 XTECH, 第 3 の量子コンピューター「冷却原子型」、30 年にも事業化へ

[7-7] John Shalf: The future of computing beyond Moore' s Law, Department of Computer Science, Lawrence Berkeley National Laboratory, 1CyclotronRoad, Berkeley, CA94720, USA (2020)

[7-8] <https://www.mri.co.jp/50th/columns/quantum/no02/>, 2020/5/15, 三菱総合研究所 50 周年記念サイト, 量子コンピューター実現の鍵はスケーラビリティの確立

第 8 章 実用化された疑似量子コンピューティング

本章では量子コンピューティングの中でも既に実用化されている例について紹介する。

まず、量子コンピューティングの分類について再度まとめる。図 8. 1 に示すように量子コンピューティングは量子ゲート方式とアニーリング方式に分けられる。量子ゲート方式は量子コンピュータのビットを量子ビットに置き換え計算する手法であり、アニーリング方式はイジングモデルを解くことで組み合わせ最適化問題を高速に解くことのみに特化した方式である。

さらにこのアニーリング方式は量子を用いた方式と疑似量子を用いた方式に分かれ、多くの企業においては疑似量子を用いた方式の計算機等が商用化されている。このアニーリング方式を用いた組み合わせ最適化問題を近似的に解くことに特化したコンピュータはアニーリングマシンやイジングマシンと呼ばれている。本章では疑似量子を用いた方式についていくつかの例を挙げてみていくことにしよう。また、本章の最後に、参照した Web サイトを一覧として載せる。

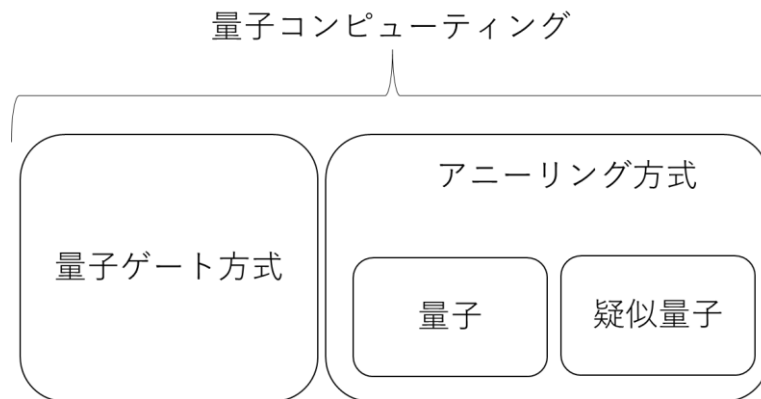


図 8.1 量子コンピューティングの内訳

8.1 アニーリングマシン・イジングマシン概説 [1]–[3]

そもそも、アニーリング方式は古くからの手法として存在はしており、古典的な方式としては、シミュレーテッドアニーリング : SA (Simulated Annealing) と呼ばれる手法がある。SA は、ある関数空間における最適化において、その系に熱揺らぎを導入することで関数空間内の最適解を探索する手法である。アニーリングは金属の熱処理である焼きなましを元にした言葉であり、収束計算を始めたばかり（焼きなましでは高温状態）では確率的に広い範囲を探索するために大きく値を振り、収束に近づいた状態（焼きなましでは低温状態）では小さく値を振ることで局所的な最適解へと収束するアルゴリズムである。図 8. 2 に SA のアルゴリズムの概念図を示す。SA は高温状態において大きく値を振ることで大域的な最適解であろう範囲を絞り込み、局所最適解を可能な限り避けることができる。しかし、SA のアルゴリズムは計算量がとても多く、実行時間に制限がある場合は探索不十分となる可能性がある。また、確率的に値を振る際に設定すべきパラメータの試行錯誤が不可欠となるため、使用にはある程度の慣れが必要であった。

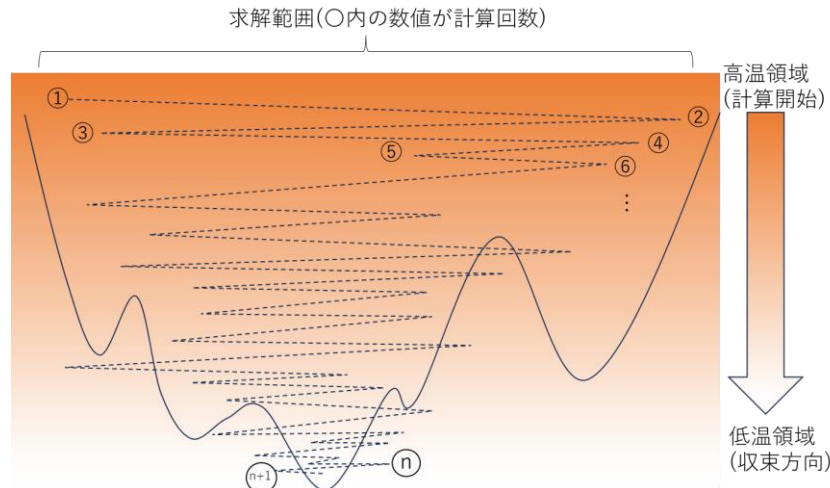


図 8.2 シミュレーテッドアニーリングアルゴリズムの概念

この SA の欠点を補うためには、実時間で考える全範囲を一斉に探索できればと考えるのが一般的といえよう。その解決策として量子コンピューティングを用いたものが量子アニーリング：QA (Quantum Annealing) である。量子アニーリングは関数空間の系に量子ゆらぎを導入することでスピン配位空間を探索する手法となる。量子揺らぎが強い領域では波動関数が非局在化し、量子揺らぎが弱くなってくることにより波動関数が徐々に局在化することで最適解に収束する。この手法は 1998 年に東京工業大学にて博士課程の学生であった門脇正史氏 (現デンソー社) と西森秀稔教授によって理論提唱され、2011 年にベンチャー企業である D-Wave Systems が商用化した。しかし、この量子アニーリングを広く実用化するには二つの課題があった。一つ目は超電導回路を用いるために巨大な冷凍機が必要となること、2 つ目は量子ビット数が少なく、本来解きたかった実用レベルの問題を扱えないということであった。

これらの量子アニーリングの課題解決のために生まれた手法が疑似量子アニーリングである。疑似量子アニーリングは古典コンピュータや半導体 CMOS 集積回路上で SA を実行し、組合せ最適化問題の近似解を求める手法であり、SA 同様に必ず最適解を導けるわけではないが、ある程度のレベルで近似解を得ることができる。

8.2 商用化済みの疑似量子アニーリングマシン [3]

では、各社から商品化製品化されている疑似量子アニーリングマシンに目を向けよう。表 1 に各社から発表されている代表的な疑似量子アニーリングマシンをまとめる。D-Wave 社は正確には疑似量子アニーリングマシンではなくハイブリッド型となるが、参考として、表に含めている。

表 1 商用化済みの疑似量子アニーリングマシン

	D-Wave社	NTT社	東芝社	NEC社	富士通社	日立	FIXSTARS
名称	Hybrid Solver Service	Coherent Ising Machine	Simulated Bifurcation	Vector Annealing	Digital Annealer	CMOS Annealing	Amplify AE
対象	QPU(超伝導量子干渉計とCPU・GPU)	レーザ・FPGA	GPU	汎用ベクトルプロセッサ	GPU	GPU	GPU
スピン数(ビット数)	100万	10万	100万～	30万	10万(10台接続可)	10万	10万～
結合方式	疎結合		全結合	全結合	全結合	全結合/疎結合	全結合
Bit階調	アナログ(5?)	不明	32	不明	64	3	64

疑似量子アニーリングの特徴は「スピン数」、「結合方式」、「ビット階調」によって表すことができる。「スピン数」は組み合わせ最適化問題をイジングモデルに定式化する際の制約であり、スピン数が多いほど大規模な問題を扱えるといえる。「Bit 階調」はパラメータの調整解像度を表し、数値が高いほど高精度に解を出せるようになる。「結合方式」には、全ての変数間に繋がりがある全結合方式と一部の変数間에만繋がりがある疎結合方式の大きく二つの分類があり、課題を疑似量子アニーリングマシンに導入するために立式したコスト関数によって好適な問題がある。このため、ユーザは解くべき問題によって、いずれの結合方式を選択するかを判断することが必要となる。図 8.3 に各結合方式の模式図を示す。全結合方式は、その名称が表す通り全てのスピン間に相互作用があるアニーリングマシンであり、異なる要素同士をすべて掛け合わせる場合に用いる。好適な問題例としてはポートフォリオ最適化、シフト最適化などである。逆に疎結合方式は一部のスピン間のみに相互作用がはたらくアニーリングマシンであり、道路などのようにネットワークが隣同士で結びつくようなモデルで用いることができる。好適な問題例としては交通渋滞削減のための交通信号制御などである。本章で紹介する多くの疑似量子アニーリングマシンは全結合に対応したモデルとなっている。

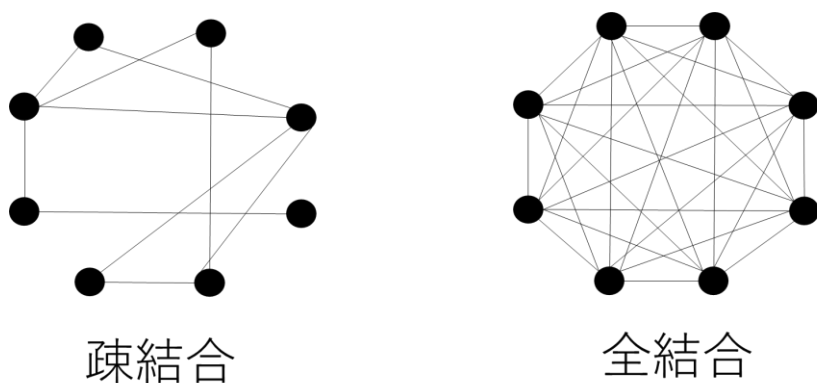


図 8.3 結合方式の模式図

では、以降で各社において商用化される疑似量子アニーリングマシンについて紹介する。

8.3 NTT 社 Coherent Ising Machine [5]–[6]

NTT 社では光パラメトリック発振器群を用いたコヒーレントイジングマシン：Coherent Ising Machine (CIM) と呼ばれる方式を開発し発表している。光パラメトリック発振器は光共振器中に位相感応

増幅器（PSA）を配置して実現する光発振器であり、この位相感応増幅器にポンプ光を入力することで、光パラメトリック増幅過程によりポンプ光位相に対して0または π 位相の成分の光がもっとも効率良く増幅される光増幅器として作用する(<https://journal.ntt.co.jp/article/10945>)。図 8.4 に CIM のモデル図を示す。

開発されたコヒーレントイジングマシンは、流す光のうち、 $0 \cdot \pi$ の位相成分を効率よく増幅する位相感応増幅器（PSA）を光ファイバーの輪のなかに配置し、PSA を 200 ピコ秒の間隔でオン・オフすることで、パルスを約 12 万個生成する。これらの光パルスは 5km の光ファイバーを数十から千周する間に相互に作用し合い、全体が最も安定する位相の組み合わせで安定することになる。この安定した組み合わせが、与えられた問題の答えとなる仕組みである。

与えられた問題の最適化計算は、中央の行列演算回路に予め解きたいイジング問題に対応するスピン間結合情報（ $N \times N$ 行列）を格納しておき、測定結果（要素数 N のベクトル）と結合行列の行列演算をすることで、次の周回における各パルスへのフィードバック情報を算出することで実行される。

CIMの特徴は、ある条件下では振幅の量子揺らぎが増幅（直交する成分は圧縮）された、いわゆるスクイーズド状態にあることで、スピンに対応する 2 つの状態が部分的に量子重ね合わせになることであり、最適解の探索に量子効果が働くものと期待されている[]。ただし CIM は、5km もの光ファイバーを要することなど、装置がある程度巨大となることになる。

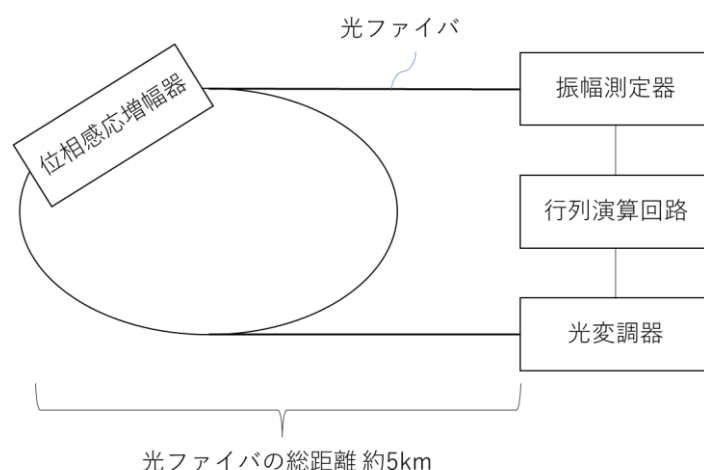


図 8.4 CIM のモデル図

8.4 東芝社 Simulated Bifurcation [7]

東芝社では、シミュレーテッドバイファケーション：Simulated Bifurcation（SB）と命名された手法を開発している。SB では、SA とは異なり、確定的なハミルトン方程式を解くために疑似乱数を利用しないため、並列化しやすいというだけでなく、疑似乱数なしで SA と同レベルの高精度な解を得ることができることが特徴である。この並列化しやすいという特徴は、PC クラスタや GPU、FPGA などを用いることで並列計算による高速化が可能となることを示唆しており、高速に問題を解きたい場合に有利に働くことが期待される。

SB では以下のようなアルゴリズムで動作しており、このアルゴリズムを核としたソリューション「SQBM

＋」では、FPGA 実装や GPU 実装など通常のサーバに実装を可能としている。また、この SQBM+では SB アルゴリズム以外でのソルバーの提供も実施しており、特定の問題を直接解くことが可能となっている。以下にその例を示す。

(<https://www.global.toshiba/jp/products-solutions/ai-iot/sbm/intro.html#contents>)

イジングソルバー：「二次制約なしバイナリ最適化問題（QUBO）」で表した組合せ最適化問題を SB アルゴリズムで解く SQBM+の基本的なソルバー。

TSP ソルバー：「巡回セールスマン問題」と呼ばれる種類の問題を QUBO で表すことなく直接解けるソルバー。

SHIFT ソルバー：各種制約条件下で従業員に各日のジョブを割り当てるようなシフトスケジューリング問題を QUBO で表すことなく直接解けるソルバー。

参考：SB アルゴリズムのフロー

Step 1. 全 x, y をランダムに初期化

Step 2. ハミルトン方程式を解く

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial y_i} = Dy_i$$

$$\frac{dy_i}{dt} = -\frac{\partial H}{\partial x_i} = -(D - p + x_i^2)x_i - cah_i + c \sum_{j=1}^N J_{i,j}x_j$$

Step 3. p を 0 から徐々に増加させシミュレーション

Step 4. x_i の符号を解として出力

8.5 NEC 社 Vector Annealing [8]–[9]

NEC 社では、もともと手掛けていた超電導回路によって構成される量子アニーリングマシンの開発と並行し、一般的なコンピュータのデジタル回路を用い、ソフトウェア処理による SA マシンを開発し、ベクターアニーリング：Vector Annealing (VA) と命名して発表している。NEC 社の量子アニーリングマシンは国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の「量子計算及びイジング計算システムの統合型研究開発」にて開発されており、リアルタイム性が重要となる無線ネットワーク最適化や自動制御系で活用することを目標として開発されている。一方、疑似量子アニーリングマシンである VA は、金融や生産計画、創薬などの大規模問題に対して活用されることを念頭に開発されており、図 8.5 に示すような求解時間と問題規模の両方向で開発が進められている。

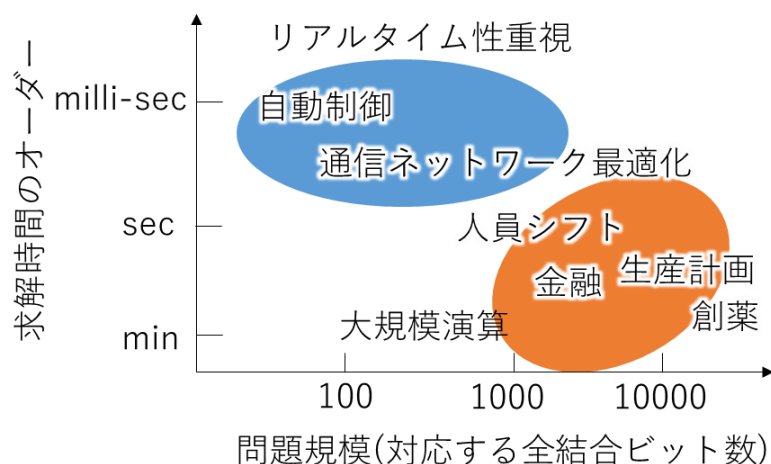


図 8.5 NEC 社の量子コンピューティング開発の方向性

また、NEC 社では、量子アニーリング処理に適した独自開発のアルゴリズムを組み込んだソフトウェアを、大容量メモリで行列計算を行うベクトル型スーパーコンピュータ「SX-Aurora TSUBASA」上で動作させ提供するようなクラウド型の疑似量子アニーリングをサービス展開している。その一方、通常のパソコンの中に搭載可能なほどの大きさで、このベクターアニーリング作成しており、GPU のように広く使える可能性を秘めている。

8.6 富士通社 Digital Annealer [10]

富士通社では開発した疑似量子アニーリングマシンを従来のデジタル技術を用いるデジタルアニーラ: Digital Annealer (DA) と命名している。DA では、近接したビット同士しか結合できないという量子ビットの制約がなく、大量のビットとビットが相互に結合し、「全結合」の状態となっており、ビット間で瞬時に情報が交換され、大きく複雑な問題でも素早く計算して解くことが可能となっている。

富士通社によると、このDAの特徴は3つあり、大規模高精度な求解を実現したことにより多様な問題に適用可能であることと、独自技術によるプロセス高速化、並列評価による計算速度の向上が挙げられている。

大規模高精度な求解という点では、10 万ビット規模の問題にまで対応しており、研究レベルでは 100 万ビットも実現しているとのことである。このビット間の結合相互作用が 64 ビットの階調ビットで完全結合することによって全てのビットが、その他のビットそれぞれに影響を与えるように設計され、高精度大規模な求解を実現している。また、プロセスの高速化では、コスト項と制約項の分離入力することで、探索途中での制約係数 α を自動調整し、チューニング作業を軽減、線形不等式制約項を直接入力可能にし、補助ビットを入力するための QUBO 化を不要としている。また状態遷移の際にどの一ビットを更新するかを判断するために全ビットについてそれぞれを反転した時の評価長を並列で計算することで最も適切なものを選び出すため遷移できずにとどまることが少なくなる。

また、富士通社では、開発した疑似量子アニーリングマシンをクラウドサービスとして Web API 化しており、イジングモデル、数式、制約条件等を入力することで問題の解を出力するサービスを展開している。このサービスは無償で提供される python 言語で使用可能なデジタルアニーラ用の PyQUBO ラ

イブラリを用いることで QUBO 生成の試用が可能となっている。

8.7 日立社 CMOS Annealing [11]–[14]

日立社では「CMOS アニーリング」と命名した疑似量子アニーリングマシンを開発し、アニーリング用のチップやハードウェアが搭載されたコンピュータ上で動作させるソリューションとセットにしたサービスを提供している。

CMOS アニーリングの特徴は大規模な組合せ最適化問題を高速に解く技術である半導体上で量子アニーリングを模した仕組みで最適解を求め、常温で操作できることと最適化計算の大規模化が可能なのが特徴である。

また日立社ではモーメンタムアニーリング：Momentum Annealing (MA) という方式の開発を進めている。もともと CMOS アニーリングにおいて用いられた従来型の SA はマルコフ連鎖モンテカルロ法に基づくため、図 8.6 に示すように全結合問題を解く場合には、繋がりを持つ変数同士を同時に更新することが不可能となり、一度に一つの変数しか更新できず、計算速度が落ちるという課題があった。

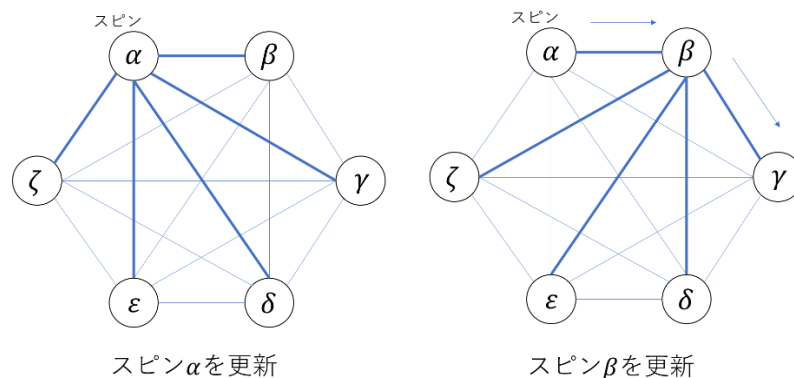


図 8.6 従来型 SA のスピン更新方法

この点を解決したアルゴリズムが MA であり、変数間の繋がりを図 8.7 のように完全二部グラフ状にし、スピンのレプリカを作ることで、図の左側に並ぶスピンは互いに繋がりを持たないため同時に更新することが可能となっている。この MA を用いた並列計算によって計算時間を短縮し、10 万スピン全結合イジングモデルの基底状態探索問題を SA に対して約 250 倍高速化可能としている。

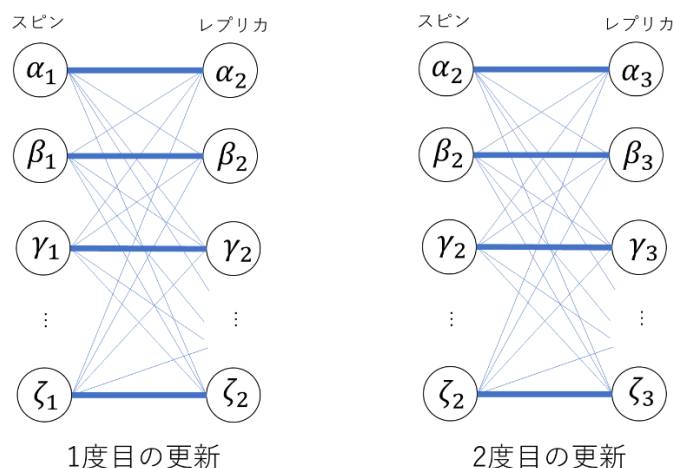


図 8.7 MA によるスピンの更新のイメージ図

8.8 FIXSTARS Amplify Annealing Engine [15]

最後に FIXSTAR 社を紹介する。FIXSTAR 社はイジングマシン向け SDK と実行環境からなるクラウド基盤を提供しており、Web 上では独自のアニーリングマシン (Amplify AE) に加え、他社のアニーリングマシン（上で紹介した富士通社、東芝社、日立社を初め、計 7 社）を利用可能にし、全ての量子アニーリング・イジングマシンや数理最適化ソルバー、ゲート式量子コンピュータに対応した量子コンピューティングプラットフォームを提供している。そのため、ユーザは図 8.8 に示すように、最適化問題を解くアプリケーションを決定した後、FIXSTARS 社が提供する SDK を用いて、ユーザのデバイス内で開発した後、FIXSTARS 社のクラウド上で使用したいアニーリングマシンを選択し、使用することができるようになる。

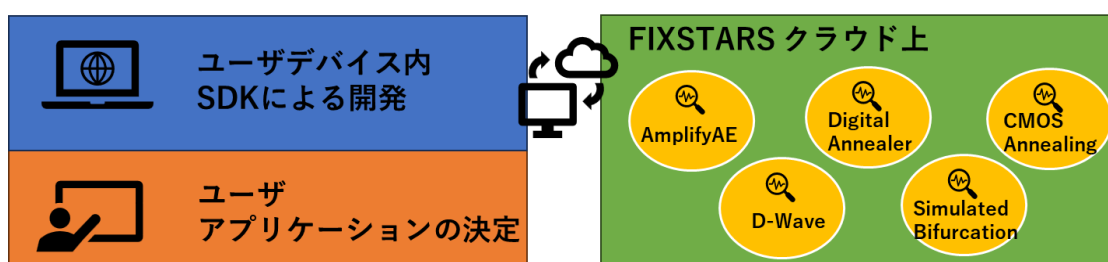


図 8.8 FIXSTARS 社 SDK を用いたアプリケーション開発の概念図

また、FIXSTAR 社の Web ページはチュートリアルが非常に豊富であり、初めてアニーリングマシンを使う場合や実際に動かしながら疑似量子コンピュータを体験してみたい人にはお勧めのコンテンツとなっている。<https://amplify.fixstars.com/ja/demo>

以下に 2023 年 7 月時点でのチュートリアルのコンテンツには、プログラミングフローの説明から始まり、もっとも単純な組み合わせ最適化問題、画像ノイズ除去、シフト最適化の一種として会議室の割り当て問題や行路最適化に用いることが可能なタクシーマッチング問題など様々なアプリケーションをサンプルコードと共に紹介している。

また、同様に疑似量子アニーリングマシンを体験、学習できる技術者の育成などを目的に無料で提供している「Annealing Cloud Web」(<https://annealing-cloud.com/ja/index.html>) を紹介する。NEDO（国立研究開発法人新エネルギー・産業技術総合開発機構）が実施する「組合せ最適化処理に向けた革新的アニーリングマシンの研究開発」の成果を使用して「量子計算およびイジング計算システムの統合型研究開発」のプロジェクトによる成果であり、FIXSTARS 社が Web を運営しており、使いながら学ぶことが可能となっている。

8.9 終わりに

本章では商用化されている疑似量子アニーリングマシンを中心に特徴などを含め紹介してきた。これまで、量子コンピュータは専用のハードが必要で、一部の限られた研究者のみが触ることができる高価な計算機であった。しかし、疑似量子アニーリングの登場により、徐々にハード面でも手が届きやすくなり、またクラウド上のハードを用いることで、API だけで計算できるようになると、ユーザには高価なハードは必要なくソフトのみで開発が可能となった。疑似量子アニーリングの用途は組合せ最適化に絞られるとしても、問題を 2 値のネットワーク最適化に落とし込むことで使用が可能となるなど、これまでに適用できないと”思い込んでいた”問題にも使用できる可能性を秘めている。

疑似量子アニーリングは、実際に使ってみることで様々な使用用途のアイデアが思いつくことだろう。ぜひ、本章で紹介したアニーリングマシンを簡単な問題から使用しながら学習し、様々な問題への適用可能性を広げていただきたい。

第 8 章の参考文献

疑似量子アニーリングマシンやイジングマシンに関して

- [8-1] Web ページ, 産総研 NEDO 量子計算及びイジング計算システムの総合型研究開発,
<https://nedo-quantum.aist.go.jp/ising-machine.html>
- [8-2] Web 記事, 「【速報】NEC 量子コンピュータ技術で設備の稼働率 15%向上・生産計画立案作業の 90%削減を達成 4 事業所に生産計画立案システムに本格導入」
<https://robotstart.info/2023/01/20/nec-smt-quantum-annealing.html>
- [8-3] Web 記事, 「イジングマシンは占い装置!? 量子アニーリングが競争の口火に」
<https://xtech.nikkei.com/atcl/nxt/column/18/01755/00002/>
- [8-4] 後藤隼人, 「イジングマシン」, 日本神経回路学会誌 Vol. 29, No. 4 (2022), 174-185
https://www.jstage.jst.go.jp/article/jnns/29/4/29_174/_pdf
※歴史から理論概要まで非常に解りやすくまとまっている論文でしたので紹介します。

NTT 社関連

- [8-5] Web 記事, 「NTT が次世代光イジングマシン、世界最大級 10 万スピンドで高速化」
<https://xtech.nikkei.com/atcl/nxt/column/18/01537/00209/?P=2>
- [8-6] NTT 技術ジャーナル記事, 「コヒーレントイジングマシンと量子アニーリングの性能比較実験」
<https://journal.ntt.co.jp/article/10945>

東芝社関連

[8-7] 「量子インスパイアード最適化ソリューション SQBM+」

<https://www.global.toshiba/jp/products-solutions/ai-iot/sbm/technology.html>

NEC 社関連

[8-8] 「量子コンピュータとシミュレーテッドアニーリング」

https://jpn.nec.com/quantum_annealing/effort/index.html

[8-9] NEC 社 NewsRoom, 「NEC、疑似量子アニーリングサービスを業界最安値で提供開始」

https://jpn.nec.com/press/202208/20220829_01.html

富士通社関連

[8-10] 「Computing as a Service Digital Annealer」

<https://www.fujitsu.com/jp/digitalannealer/>

日立社関連

[8-11] 内閣府 量子技術の実用化推進ワーキンググループ（第3回）, 「日立の量子コンピュータ研究開発戦略」

https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/3kai.html

https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/siryos2-2_1.pdf

[8-12] Web 記事, 「日立の次世代技術「CMOS アニーリング」実用化へ(1) 量子コンピュータを疑似的に再現 アニーリングと組合せ最適化とは【入門編】」

<https://robotstart.info/2021/01/31/hitachi-annealing-01.html>

[8-13] AnealingCloudWeb 解説記事, 「CMOS アニーリングマシン 全結合問題への拡張」

<https://annealing-cloud.com/ja/about/momentum-annealing.html>

[8-14] 日立評論, 「CMOS アニーリングの顧客適用に向けた量子コンピュータ技術の応用」, 2020 vol. 102 No. 3

<https://www.hitachihyoron.com/jp/archive/2020s/2020/03/03b09/index.html>

FIXSTARS 社関連

[8-15] 「量子コンピューティングプラットフォーム」 : <https://amplify.fixstars.com/ja/>

ノート A 複素ベクトル空間

参考文献：中山 茂，量子アルゴリズム，技報堂出版，第 1 版，(2014)，p. 113

1. ベクトル空間とヒルベルト空間

ヒルベルト空間とは内積が定義された完備な複素ベクトル空間である。完備性とはコーシー列が収束する空間である。コーシー列とはその任意の項同士の距離が限りなく小さくなるような列のことを指す。

複素ベクトル $|x\rangle, |y\rangle, |z\rangle, \dots$ で作られる空間を考える。ベクトル空間とはベクトルの和とスカラー倍が定義された空間で次の性質がある。

- ・交換法則が成立： $|x\rangle + |y\rangle = |y\rangle + |x\rangle$
- ・結合法則が成立： $(|x\rangle + |y\rangle) + |z\rangle = |x\rangle + (|y\rangle + |z\rangle)$
- ・ゼロの存在： $|x\rangle + |\emptyset\rangle = |x\rangle$
- ・逆の存在： $|x\rangle + |-x\rangle = |\emptyset\rangle$
- ・分配則： $a(|x\rangle + |y\rangle) = a|x\rangle + a|y\rangle, (a+b)|x\rangle = a|x\rangle + b|x\rangle$

2. 2次元複素ベクトル空間

- ・ベクトル $|x\rangle$ が 2 個の複素数からなるとき次のよう列ベクトルで表す： $|x\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$
- ・随伴（共役転置）行列は行ベクトルで： $\langle x| = (|x\rangle)^\dagger = (x_0^* \ x_1^*)$
 $\langle x|$ と $|x\rangle$ はそれぞれブラベクトル，ケットベクトルと呼ばれる。
- ・ベクトルの和とスカラー乗： $\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} + \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 + y_0 \\ x_1 + y_1 \end{pmatrix}, \alpha \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} \alpha x_0 \\ \alpha x_1 \end{pmatrix}$
- ・2次元複素ベクトルの基底： $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \langle 0| = (1 \ 0), \langle 1| = (0 \ 1)$

$$|x\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = x_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + x_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = x_0 |0\rangle + x_1 |1\rangle; \quad \langle x| = (x_0 \ x_1) = x_0 \langle 0| + x_1 \langle 1|$$

3. ベクトルの内積と外積

(1) ベクトルの内積

・定義：

$$|x\rangle = x_0 |0\rangle + x_1 |1\rangle, \quad |y\rangle = y_0 |0\rangle + y_1 |1\rangle \\ \rightarrow \langle x|y\rangle = (|x\rangle)^\dagger |y\rangle = (x_0^* \ x_1^*) \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = x_0^* y_0 + x_1^* y_1$$

・性質：

$$\square \langle x|y\rangle = x_0^* y_0 + x_1^* y_1 = (x_0 y_0^* + x_1 y_1^*)^* = (\langle y|x\rangle)^*$$

$$\square \langle x|(a|y\rangle + b|z\rangle) = a\langle x|y\rangle + b\langle x|z\rangle$$

$$\square \langle x|x\rangle = |x_0|^2 + |x_1|^2 \geq 0$$

$$\square \langle x|x\rangle = 0 \Leftrightarrow |x\rangle = |\emptyset\rangle$$

$$\bullet \text{ シュワルツの不等式 : } |\langle x|y\rangle|^2 \leq \langle x|x\rangle \langle y|y\rangle \leq \|x\|^2 \|y\|^2$$

$$\bullet \text{ 三角不等式 : } \|x+y\|^2 \leq \|x\|^2 + \|y\|^2$$

(2) ベクトルの外積

□ 定義 :

$$|x\rangle\langle y| = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \begin{pmatrix} y_0^* & y_1^* \end{pmatrix} = \begin{pmatrix} x_0 y_0^* & x_0 y_1^* \\ x_1 y_0^* & x_1 y_1^* \end{pmatrix}$$

内積は数になるが, 外積は行列 (演算子) になる!

• トレース :

$$\text{Tr}(|x\rangle\langle y|) = x_0 y_0^* + x_1 y_1^* = \begin{pmatrix} y_0^* & y_1^* \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \langle y|x\rangle$$

$$\bullet \text{ ケットベクトルに作用 } \rightarrow (|x\rangle\langle y|)|y'\rangle = |x\rangle\langle y|y'\rangle = \langle y|y'\rangle |x\rangle$$

$$\bullet \text{ ブラベクトルに作用 } \rightarrow \langle x'|(|x\rangle\langle y|) = \langle x'|x\rangle \langle y|$$

• 外積の展開 :

$$\begin{aligned} |x\rangle\langle y| &= \begin{pmatrix} x_0 y_0^* & x_0 y_1^* \\ x_1 y_0^* & x_1 y_1^* \end{pmatrix} = x_0 y_0^* \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + x_0 y_1^* \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + x_1 y_0^* \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + x_1 y_1^* \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ &= x_0 y_0^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} + x_0 y_1^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} + x_1 y_0^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} + x_1 y_1^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \\ &= x_0 y_0^* |0\rangle\langle 0| + x_0 y_1^* |0\rangle\langle 1| + x_1 y_0^* |1\rangle\langle 0| + x_1 y_1^* |1\rangle\langle 1| = \sum_{i=0}^1 \sum_{j=0}^1 x_i y_j^* |i\rangle\langle j| = \sum_{i=0}^1 \sum_{j=0}^1 \alpha_{ij} |i\rangle\langle j| \end{aligned}$$

4. 線形作用素

$$\bullet \text{ 定義 : } T(a|x\rangle + b|y\rangle) = aT|x\rangle + bT|y\rangle$$

$$2 \text{ 次元複素ベクトル空間の場合には : } T = \begin{pmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{pmatrix}$$

$$\bullet \text{ 転置 (エルミート) 共役 : } T^\dagger = (T^T)^* = \begin{pmatrix} t_{00}^* & t_{10}^* \\ t_{01}^* & t_{11}^* \end{pmatrix}$$

$$\text{自己共役 (エルミート) 性 : } T = T^\dagger$$

- ・ 恒等作用素 : $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- ・ ユニタリー作用素 : $UU^\dagger = U^\dagger U = I$
- ・ ユニタリー作用素 = 直交作用素 : $[U, U^\dagger] = UU^\dagger - U^\dagger U = 0$

5. 線形作用素の固有値と固有ベクトル

- ・ 定義 (固有値と固有ベクトル) : $T|x\rangle = \lambda|x\rangle$
- ・ エルミート ($T = T^\dagger$) で, ユニタリー ($TT^\dagger = I$) な作用素の固有値は実数で大きさは 1 :
 $|x\rangle = I|x\rangle = TT^\dagger|x\rangle = TT|x\rangle = \lambda^2|x\rangle \rightarrow 1 = \lambda^2$
- ・ 固有値方程式 : $T|x\rangle = \lambda|x\rangle \rightarrow (T - \lambda I)|x\rangle = 0 \rightarrow |T - \lambda I| = 0$
- ・ 2次元複素ベクトルの場合の固有値方程式 :

$$\begin{vmatrix} t_{00} - \lambda & t_{01} \\ t_{10} & t_{11} - \lambda \end{vmatrix} = 0 \rightarrow \lambda^2 - (t_{00} + t_{11})\lambda + t_{00}t_{11} - t_{01}t_{10} = 0$$

- ・ パウリ行列 X の固有値方程式 : $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \rightarrow \begin{vmatrix} -\lambda & 1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 - 1 = 0 \rightarrow \lambda = \pm 1$
- ・ パウリ行列 X の固有ベクトル : $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \pm \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \rightarrow \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix} \rightarrow |x\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$

$$\text{検算 : } \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\text{性質 : } X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \quad X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

6. ベクトルのテンソル積

- ・ 2 個の複素ベクトル : $|x\rangle = x_0|0\rangle + x_1|1\rangle = (x_0 \ x_1)^T$, $|y\rangle = y_0|0\rangle + y_1|1\rangle = (y_0 \ y_1)^T$
- ・ テンソル積の定義 :

$$\begin{aligned} |x\rangle \otimes |y\rangle &= (x_0|0\rangle + x_1|1\rangle) \otimes (y_0|0\rangle + y_1|1\rangle) \\ &= x_0y_0|0\rangle \otimes |0\rangle + x_0y_1|0\rangle \otimes |1\rangle + x_1y_0|1\rangle \otimes |0\rangle + x_1y_1|1\rangle \otimes |1\rangle \\ &= x_0y_0|00\rangle + x_0y_1|01\rangle + x_1y_0|10\rangle + x_1y_1|11\rangle \end{aligned}$$

$$10 \text{ 進表示による書き直し : } |x\rangle \otimes |y\rangle = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + c_3|3\rangle$$

新しい基底 :

$$|0\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|2\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |3\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

ノート B 量子ゲート問題の基礎

参考文献：中山 茂，量子アルゴリズム，技報堂出版，第 1 版，(2014)，p. 113

1. 量子計算の原点

1.1 量子ビット (quantum bit, qubit)

・量子ビットは量子計算の基本的な情報単位である。古典的なビットは、0 または 1 の 2 つの状態を持つが、量子ビットは「重ね合わせ状態」という、0 と 1 の 2 つの状態が同時に存在する量子力学特有の状態である。

・観測するまでどちらの状態とも言えないが、観測すれば、0 または 1 のどちらかの状態になる。

・式で書くと次式のように書ける： $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ ， $|\alpha|^2 + |\beta|^2 = 1$

α と β は複素数で $|\alpha|^2$ と $|\beta|^2$ は、観測時に $|\psi\rangle$ が $|0\rangle$ か $|1\rangle$ になる確率である。

・光の偏光，電子のスピン，原子のエネルギー準位，ジョセフソン素子の電流回転方向などで物理的に実現できる。

1.2 重ね合わせ状態

・1 量子ビットの重ね合わせ状態 $|\psi\rangle$ は、 $|0\rangle$ と $|1\rangle$ を基底とする 2 次元列ベクトルとして表現可能：

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

・2 量子ビットの重ね合わせ状態は 4 次元列ベクトル：

$$\begin{aligned} |\psi\rangle &= (\alpha'|0\rangle + \beta'|1\rangle)(\gamma'|0\rangle + \delta'|1\rangle) = \alpha|0\rangle|0\rangle + \beta|0\rangle|1\rangle + \gamma|1\rangle|0\rangle + \delta|1\rangle|1\rangle \\ &= \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \gamma \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \delta \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad |0\rangle = |0\rangle|0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle = |0\rangle|1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\ &\quad |2\rangle = |1\rangle|0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |3\rangle = |1\rangle|1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

・ビット表現の違い

(1) 古典ビット…点

0 または 1 しか許されない。

(2) 確率ビット…線

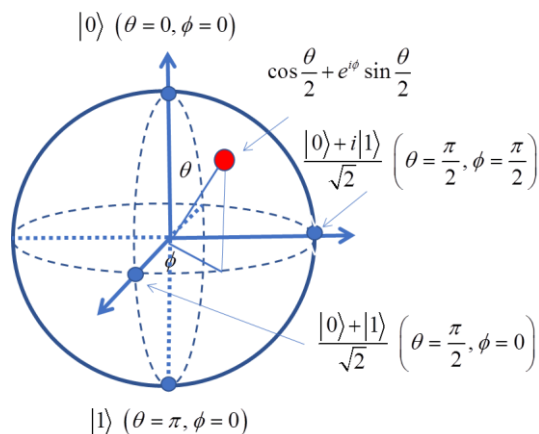
確率 P ($0 \leq P \leq 1$) で 0, $1-P$ で 1 になる。

(3) 量子ビット…面

$$\text{ブロッホ球面} : |\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \phi < 2\pi$$

1 量子ビットの状態がブロッホ球面の 1 点と過不足なく 1 : 1 に対応する。

・ブロッホ球



ブロッホ球 : $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \phi < 2\pi$

1.3 量子論理ゲートでの万能ゲート

- ・ 万能古典ゲートは, 2 入力 1 出力の NAND である.



古典ゲート NAND

- ・ 万能量子論理ゲートは, 1 入力 1 出力の万能ユニタリーゲートと 2 入力 2 出力の制御 NOT (CNOT) ゲートの組み合わせでできる.



量子ゲート CNOT

- ・ 万能ユニタリーゲートとは, 1 量子ビットのゲートで, 入力状態をブロッホ球上のどこにでも回転して移動できるものである.
- ・ 1 量子ビットの万能ゲートについては後述する.

2. 情報と熱との関係 (ランダウアーの原理)

- ・ 情報エントロピー : 系の取る状態 = 等確率の N 状態とすると $\rightarrow S = k_B \ln N$
- ・ 2 入力 1 出力の古典ゲート $\rightarrow \begin{cases} S_{in} = k_B \ln N_{in} = k_B \ln 2^2 = 2k_B \ln 2 \\ S_{out} = k_B \ln N_{out} = k_B \ln 2 = k_B \ln 2 \end{cases} \quad \Delta S = S_{in} - S_{out}$
- ・ エントロピー減少 (情報ロス) \rightarrow 発熱 $\left(\Delta S = \frac{\Delta Q}{T} \right)$
- ・ 量子ゲートは入力と出力のビット数が等しい \rightarrow エントロピーロスなし
従って, 発熱なし!

3. 1 量子ビットの量子論理ゲート

・パウリのXゲート（ビット反転）： $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

・パウリのYゲート（ビット位相反転）： $Y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$

・パウリのZゲート（位相反転）： $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

・アダマールゲート： $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

・回転変換： $P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$

・位相シフト変換： $R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$

3.1 1量子ビットの万能ゲート

・1量子ビットの万能ユニタリーゲートは、HゲートとPゲートでできる。

$$|0\rangle \xrightarrow{H} \xrightarrow{P_\theta} \xrightarrow{H} \xrightarrow{P_{\phi+\pi/2}} |\psi\rangle$$

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$

∴)

$$\begin{aligned} |0\rangle &\xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \xrightarrow{P_\theta} \frac{|0\rangle + e^{i\theta}|1\rangle}{\sqrt{2}} \xrightarrow{H} \frac{(|0\rangle + |1\rangle) + e^{i\theta}(|0\rangle - |1\rangle)}{2} = \frac{(1+e^{i\theta})|0\rangle + (1-e^{i\theta})|1\rangle}{2} \\ &= e^{i\theta/2} \frac{(e^{-i\theta/2} + e^{i\theta/2})|0\rangle + (e^{-i\theta/2} - e^{i\theta/2})|1\rangle}{2} = e^{i\theta/2} \left(\cos \frac{\theta}{2} |0\rangle - i \sin \frac{\theta}{2} |1\rangle \right) \\ &\xrightarrow{P_{\pi/2+\phi}} e^{i\theta/2} \left(\cos \frac{\theta}{2} |0\rangle - i e^{i(\pi/2+\phi)} \sin \frac{\theta}{2} |1\rangle \right) = e^{i\theta/2} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) \sim \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \end{aligned}$$

・グローバルな位相は $|e^{i\theta/2}|^2 = 1$ なので、確率に影響を与えないので無視できる。故にブロッホ球面上の任意の点は $|0\rangle$ から、上の図で定義される複合ゲートで到達できる。

3.2 平方根ゲート

同一の二つのゲートの積がユニタリーゲート U に等しいなら、そのゲートは U の平方根であると言える。

$$\sqrt{U}(\sqrt{U}|\psi\rangle) = (\sqrt{U}\sqrt{U})|\psi\rangle = U|\psi\rangle$$

例) $NOT = X$ の平方根

$$\sqrt{NOT} = \begin{pmatrix} a & b \\ b & a \end{pmatrix} \rightarrow \begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} a & b \\ b & a \end{pmatrix} = NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$a^2 + b^2 = 0, 2ab = 1 \rightarrow b = ia \rightarrow 2ia^2 = 1 \rightarrow a^2 = -\frac{i}{2} \rightarrow a = \pm i \frac{\sqrt{i}}{\sqrt{2}}$$

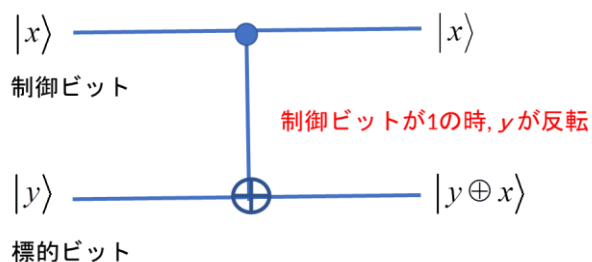
$$\rightarrow a = \pm i \frac{1}{\sqrt{2}} \frac{1+i}{\sqrt{2}} \rightarrow a = \pm i \frac{1+i}{2} = \mp \frac{1-i}{2}, b = \mp \frac{1+i}{2}$$

$$\therefore \sqrt{NOT} = \frac{1}{2} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix}, \quad \sqrt{NOT} \sqrt{NOT} = \frac{1}{4} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = NOT$$

4. 2 量子ビットの量子論理ゲート

4.1 制御 NOT ゲート

- ・ 制御 NOT ゲートは、量子ゲート問題で最も重要な 2 量子ビットゲートの一つ。



制御 NOT ゲート

$$|0\rangle|0\rangle \xrightarrow{CNOT} |0\rangle|0 \oplus 0\rangle = |0\rangle|0\rangle,$$

$$|0\rangle|1\rangle \xrightarrow{CNOT} |0\rangle|1 \oplus 0\rangle = |0\rangle|1\rangle,$$

$$|1\rangle|0\rangle \xrightarrow{CNOT} |1\rangle|0 \oplus 1\rangle = |1\rangle|1\rangle,$$

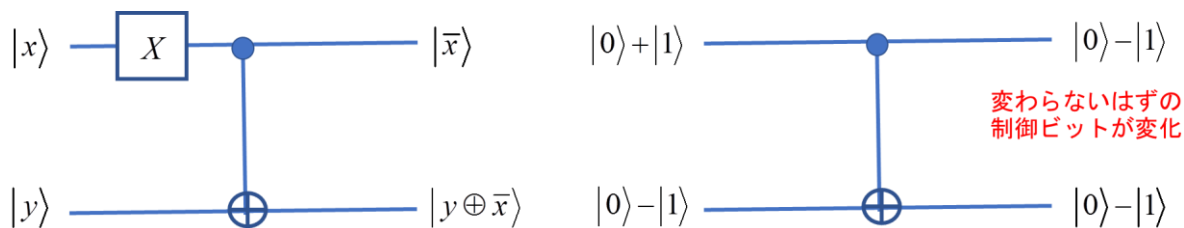
$$|1\rangle|1\rangle \xrightarrow{CNOT} |1\rangle|1 \oplus 1\rangle = |1\rangle|0\rangle$$

	input	output
$CNOT =$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{matrix} 00\rangle & 00\rangle \\ 01\rangle & 01\rangle \\ 10\rangle & 11\rangle \\ 11\rangle & 10\rangle \end{matrix}$
		\rightarrow

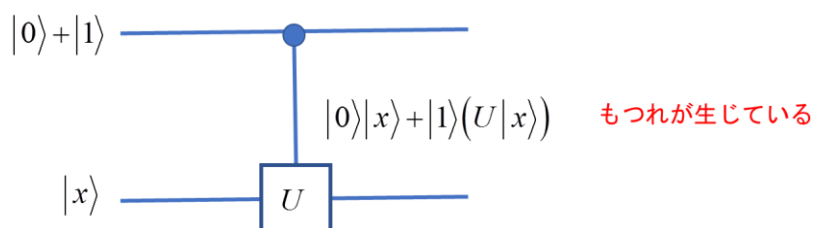
- ・ もつれ状態の生成

$$(|0\rangle + |1\rangle)|0\rangle = |0\rangle|0\rangle + |1\rangle|0\rangle \xrightarrow{CNOT} |0\rangle|0\rangle + |1\rangle|1\rangle \quad \text{1 ビットと 2 ビットに分離不能になっている。}$$

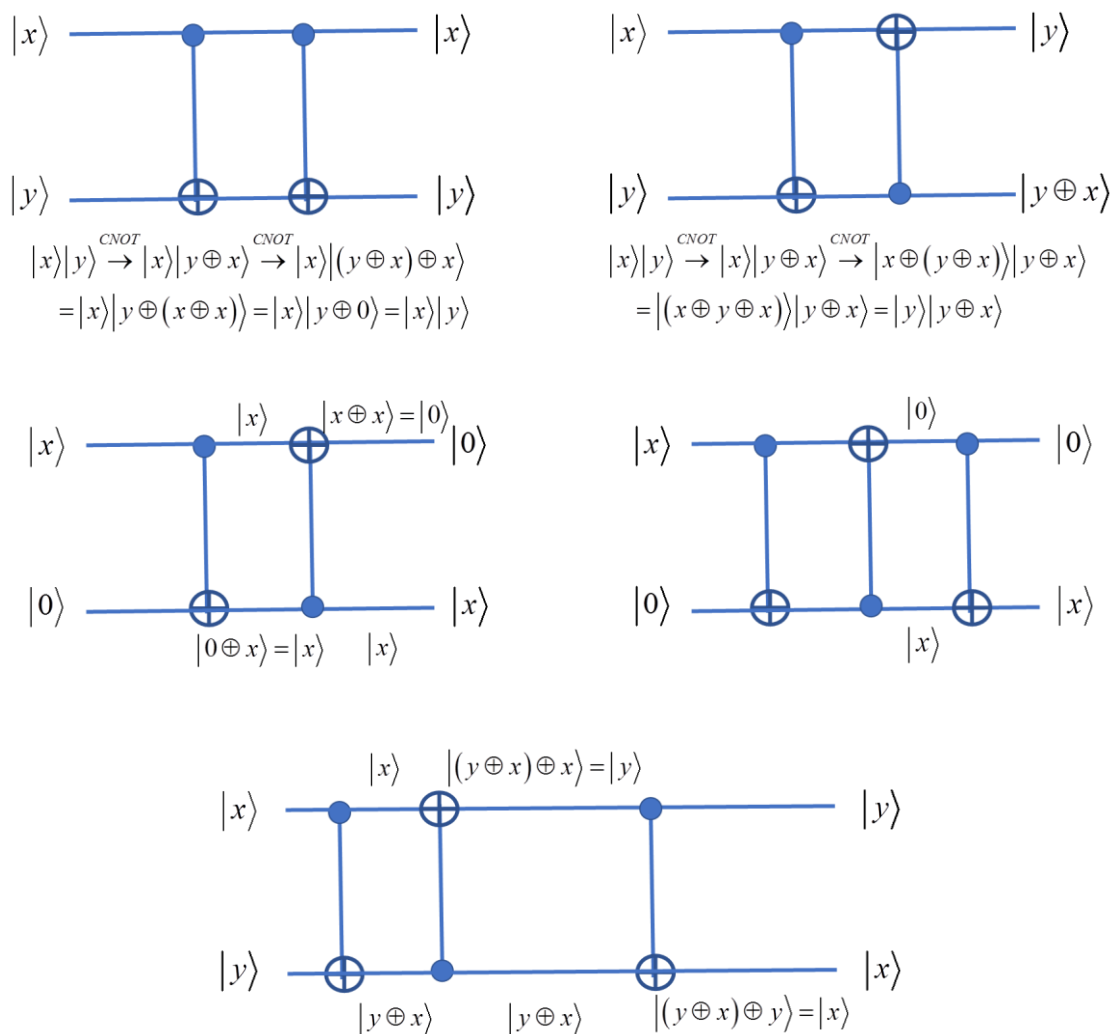
- ・ 制御 NOT ゲートの使用例



4.2 制御 U ゲート

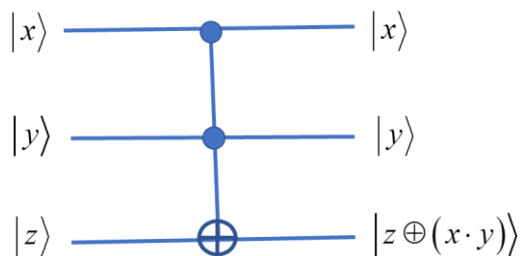


4.3 制御 NOT ゲートの組み合わせ

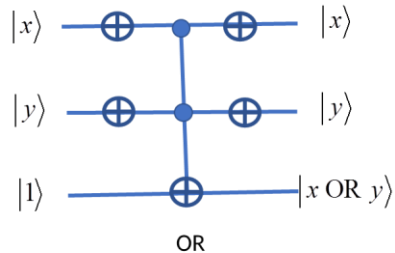
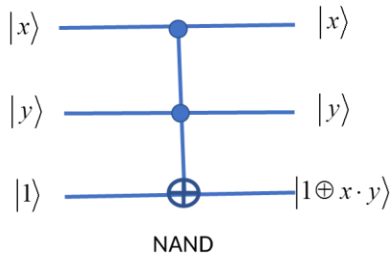
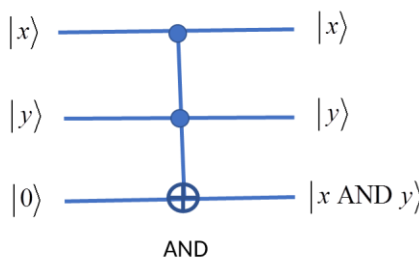
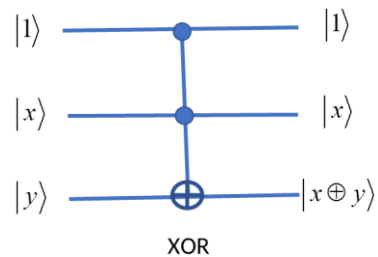
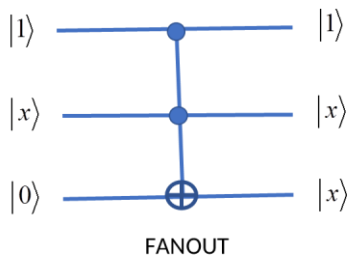
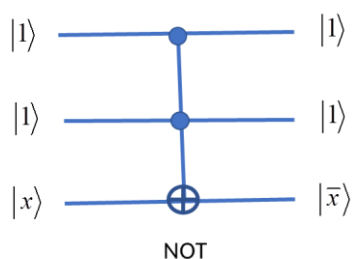


5. 3量子ビットの量子論理ゲート

- ・トフォリゲート：制御・制御 NOT (CCNOT) と呼ぶ。 $x=y=1$ の時に $|z\rangle \rightarrow |\bar{z}\rangle$, 他は素通り。

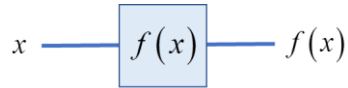


								input	output
TOFFOLI =	1	0	0	0	0	0	0	$ 000\rangle$	$ 000\rangle$
	0	1	0	0	0	0	0	$ 001\rangle$	$ 001\rangle$
	0	0	1	0	0	0	0	$ 010\rangle$	$ 010\rangle$
	0	0	0	1	0	0	0	$ 011\rangle$	$ 011\rangle$
	0	0	0	0	1	0	0	$ 100\rangle$	$ 100\rangle$
	0	0	0	0	0	1	0	$ 101\rangle$	$ 101\rangle$
	0	0	0	0	0	0	1	$ 110\rangle$	$ 111\rangle$
	0	0	0	0	0	0	1	$ 111\rangle$	$ 110\rangle$
	0	0	0	0	0	0	0		

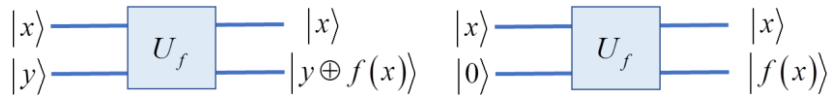


6. オラクル計算

- ・関数値を計算する仮想的なゲート



古典的なオラクル（不可逆）

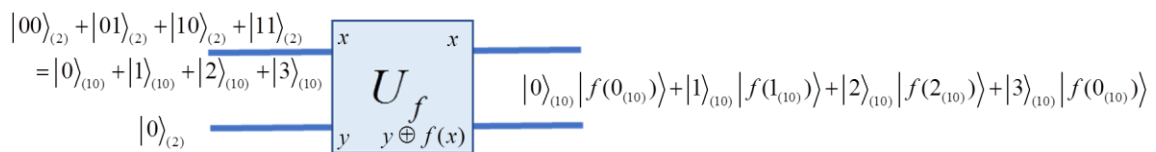
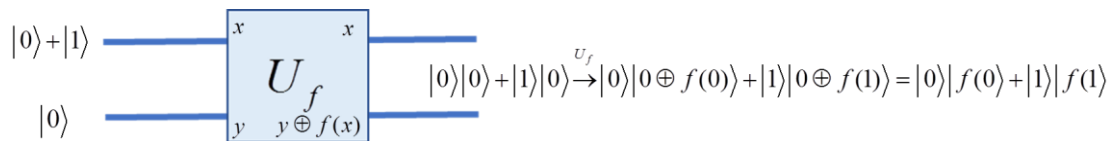


量子オラクル（可逆）

量子オラクル（可逆）

・可逆にするために 2 量子ビットのゲート

$$\begin{aligned}
 \text{順方向: } & |x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle \\
 & |x\rangle|y \oplus f(x)\rangle \xrightarrow{U_f} |x\rangle|(y \oplus f(x)) \oplus f(x)\rangle \\
 \text{逆方向: } & = |x\rangle|y \oplus (f(x) \oplus f(x))\rangle = |x\rangle|y \oplus 0\rangle \\
 & = |x\rangle|y\rangle
 \end{aligned}$$



$$\begin{aligned}
 & \left(|0\rangle_{(10)} + |1\rangle_{(10)} + |2\rangle_{(10)} + |3\rangle_{(10)} \right) |0\rangle_{(2)} \quad \longrightarrow \quad |0\rangle_{(10)} |0_{(2)} \oplus f(0_{(10)})\rangle + |1\rangle_{(10)} |0_{(2)} \oplus f(0_{(10)})\rangle \\
 & \quad + |2\rangle_{(10)} |0_{(2)} \oplus f(0_{(10)})\rangle + |3\rangle_{(10)} |0_{(2)} \oplus f(0_{(10)})\rangle \\
 & = |0\rangle_{(10)} |f(0_{(10)})\rangle + |1\rangle_{(10)} |f(1_{(10)})\rangle \\
 & \quad + |2\rangle_{(10)} |f(2_{(10)})\rangle + |3\rangle_{(10)} |f(3_{(10)})\rangle
 \end{aligned}$$

編著者 一色 浩（１～６章）

工学博士。数理解析研究所(大阪) 代表。

1965 年 東京大学工学部卒業

1972 年 東京大学大学院博士課程終了

韓国ソウル大学講師

1973 年 日立造船株式会社 技術研究所を経て

2001 年 大阪に数理解析研究所を設立

2007-2011 年 韓国蔚山大学碩座教授

2016 年 測位技術振興会顧問。現在に至る。

著者 毛利 篤史（７章）

博士(工学)。三菱電機(株)。

1989 年 大阪府立大学大学院工学研究科修士課程修了

1989 年 三菱電機(株)LSI 研究所に入社

2002 年 三菱電機(株)三田製作所へ異動

2017 年 立命館大学大学院理工学研究科博士課程修了

2018 年 測位技術振興会理事。現在に至る。

著者 板東 幹雄（８章）

博士(工学)。(株)日立製作所。

2004 年 京都大学大学院情報学研究科修士課程修了

2004 年 (株)日立製作所日立研究所(当時)に入社

2018 年 測位技術振興会理事

2019 年 大阪大学大学院工学研究科博士課程修了

現在に至る。

古典力学から量子力学へ そして量子計算へ

2023年11月1日 1.0 版発行

オープン価格

編著者 一色 浩

発行者 測位技術振興会

装填・電子化 舞浜書房

Published in Japan



測位技術振興会