IMES DISCUSSION PAPER SERIES

量子コンピュータによる暗号解読アルゴリズム：隠れ部分群問題から見た素因数分解問題と離散対数問題

國廣 昇（くにひろ のぼる）

Discussion Paper No. 2026-J-1

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

日本銀行金融研究所が刊行している論文等はホームページからダウンロードできます。
https://www.imes.boj.or.jp

IMES Discussion Paper Series 2026-J-1
2026 年 1 月

量子コンピュータによる暗号解読アルゴリズム：隠れ部分群問題から見た素因数分解問題と離散対数問題

國廣　昇（くにひろ　のぼる）*

要　　旨

現在広く利用されている公開鍵暗号は、素因数分解問題や離散対数問題の求解の難しさに安全性を依拠している。暗号に使用されるこれらの問題は、現在のコンピュータでは計算が困難であるものの、大規模な量子コンピュータが実現すれば容易に求解できることがわかっている。素因数分解問題や離散対数問題を求解できる量子コンピュータの規模は、求解に必要となる計算量に依存することから、暗号が解読され得る時期を予測するには、量子コンピュータのハードウェア開発に関する動向に加えて、求解の方法（アルゴリズム）の効率化にかかる研究の進展にも注目する必要がある。1994 年に初めて素因数分解等を求解するアルゴリズムが提案されて以降、量子コンピュータによって容易に求解可能となる問題に関する研究は進展しており、その求解可能となる問題の範囲は、素因数分解問題や離散対数問題も含む可換群の隠れ部分群問題と呼ばれる問題にまで拡大している。こうした研究の進展を受け、これまでの想定より小規模な量子コンピュータで素因数分解問題の求解が可能となることもわかってきている。本稿では、こうした研究の進展によって明らかとなった、可換群の隠れ部分群問題、および、素因数分解問題と離散対数問題の関係について解説するとともに、これらの問題が量子コンピュータによって容易に求解できる仕組みを理解するため、そのアルゴリズムを解説する。

キーワード：隠れ部分群問題、素因数分解問題、耐量子計算機暗号、離散対数問題、量子アルゴリズム、量子コンピュータ

JEL classification: L86、L96、Z00

* 筑波大学システム情報系教授（E-mail: kunihiro@cs.tsukuba.ac.jp）

本稿は、筆者が日本銀行金融研究所客員研究員の期間に行った研究をまとめたものである。本稿の作成に当たっては、高島克幸教授（早稲田大学）と高安敦准教授（東京大学）から有益なコメントを頂戴した。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行あるいは筑波大学の公式見解を示すものではない。また、ありうべき誤りはすべて筆者個人に属する。

目次

第 1 章

はじめに

暗号技術は，情報セキュリティの基盤である．暗号技術に脆弱性があれば，期待通りの情報セキュリティを確保することができない．そのため，情報セキュリティの確保には，安全であることが数学的に証明された暗号技術の使用が必須となる．RSA 暗号 [23] やデジタル署名技術 ECDSA [9] といった現在広く利用されている暗号技術[*1]（以下，従来暗号技術）の多くは，素因数分解問題や離散対数問題[*2]における求解の難しさに安全性の根拠を置く．つまり，最も効率のよいアルゴリズムを用いたとしても，現在のコンピュータではその計算に膨大な時間がかかることから，従来暗号技術は安全といわれる[*3]．

しかし，1994 年，量子コンピュータがあれば，素因数分解問題や離散対数問題が容易に求解できることがショアによって示された [24]．現行のコンピュータ（量子コンピュータに対して古典コンピュータと呼ばれる）が 0 と 1 どちらかの状態しか保持できない古典ビットを用いて計算を行っているのに対し，量子コンピュータは，量子力学の原理に基づき，0 と 1 両方の状態を同時にとることができる量子ビットを使用することから，複数の状態を同時に扱うことができるという特徴がある．ショアは，こうした量子力学の原理を利用することで，素因数分解問題と離散対数問題を容易に求解する方法を理論的に示した．

もっとも，従来暗号技術が安全性の根拠としている大きさの素因数分解問題や離散対数

[*1] これら暗号技術の詳細は付録 B を参照されたい．

[*2] 離散対数問題とは，3 つの整数 g, h, p に対して，$h \equiv g^x \pmod{p}$ となる x を求める問題である．なお，ECDSA は楕円離散対数問題の求解の難しさに安全性の根拠をおくデジタル署名方式であるが，離散対数問題と楕円離散対数問題は問題を定義する群の違いによるものであることから，本稿では離散対数問題をベースに整理を進めることとする．なお，群とは，ある条件を満たす演算が与えられた集合をいう．正確な定義などは，付録 A.1 を参照されたい．

[*3] こうした安全性は計算量的安全性と呼ばれる．

問題について，これらの求解に必要な量子コンピュータが実現する見通し時期は専門家によってさまざまであり，いつ，従来暗号技術が安全でなくなるのかを明らかにすることは難しい．現時点では，量子コンピュータが素因数分解できているのは 5 ビットの合成数[*4]までであり [12]，RSA 暗号に用いられている 2048 ビットの合成数との間には大きな乖離がある．しかしながら，量子コンピュータ技術は驚異的な速度で進展しており，従来暗号技術の将来的な危殆化のリスクに備えた対策を適切に実施していくことが必要であろう．

こうした課題への対応には，主に以下の 2 つがある．

(1) 量子コンピュータにより，従来暗号技術が安全性の根拠としている大きさの素因数分解問題や離散対数問題の求解が可能となる時期を予測する．
(2) 量子コンピュータに対して耐性のある暗号技術（以下，耐量子計算機暗号）を開発し，標準化する．

上記対応により，従来暗号技術の利用者は，(1) による予測時期を見通して，(2) で開発・標準化された耐量子計算機暗号に移行していくことが想定される．すでに海外では耐量子計算機暗号への移行に向けた動きが進んでおり，国内においても検討が進められている [5]．

(1) を行うには，まず，量子コンピュータによる求解アルゴリズム（以下，量子アルゴリズム）に必要なリソース（計算量や量子ビット[*5]数）を算出する必要がある．量子アルゴリズムに求められるリソースがわかれば，その実装に要請される量子コンピュータの規模や性能が明らかとなり，求解可能な時期の予測につながる[*6]．ショアによって素因数分解問題と離散対数問題の量子アルゴリズムが示された 1994 年以降，研究の進展により，量子コンピュータで容易に求解できる問題は素因数分解問題や離散対数問題を含む可換群[*7]の隠れ部分群問題[*8]と呼ばれる問題全般にまで拡張できることが明らかとなっている．すなわち，可換群の隠れ部分群問題に含まれる問題であれば，量子コンピュータによって容易に求解できることがわかっている．さらに，求解可能な具体的問題の洗い出しを目的と

[*4] 素因数分解された合成数は $21 = 3 \times 7$ である．なお，21 は 5 ビットであり，2 進数で表すと 10101 である．

[*5] 量子コンピューティングにおける情報の基本単位．古典コンピュータにおけるビット（0 か 1）に相当する．

[*6] 量子アルゴリズムに求められるリソースが大きくなればなるほど，その実装に必要な量子コンピュータの規模は大きくなる．

[*7] 演算の順序が交換可能である場合（例えば，演算「$\cdot$」に対して，$a \cdot b = b \cdot a$ が成り立つ場合），その群は可換群と呼ばれる．詳細は付録 A.1 を参照のこと．

[*8] 部分群とは，ある群の部分集合であって，それ自身も群であるものをいう．また，隠れ部分群問題とは，群の中に隠れている部分群を探す問題である．詳しくは，3 章および付録 C を参照．

した研究の対象は，非可換群の隠れ部分群問題も含めた隠れ部分群問題全般に広がっている．また，耐量子計算機暗号は，量子コンピュータであっても容易に求解できない数学的問題をベースとする必要があるため，耐量子計算機暗号の安全性評価の観点からも隠れ部分群問題に関する研究が重要となっている[*9]．

隠れ部分群問題とは，群の中に隠れている部分群を探す問題である．よりわかりやすく説明すると，関数 f の出力値が同じになる元 x と y（$f(x) = f(y)$）があるとき，その元の差 $x - y$ が形成する部分群を求める問題である（図 1 参照）．

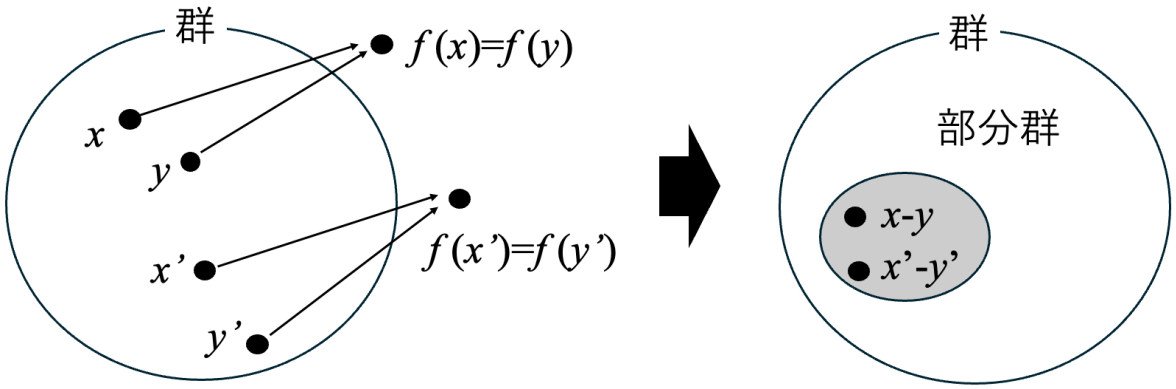

図 1　隠れ部分群

なお，量子コンピュータであっても，引き続き求解が難しい問題も存在する．例えば，共通鍵暗号の秘密鍵を探索する量子アルゴリズムは存在するものの，その計算量は膨大であることから，大規模な量子コンピュータが実現したとしても，共通鍵暗号の安全性に大きな影響はないといわれている．

(2) に関しては，現在，世界各国で耐量子計算機暗号の研究開発が進められている．米国国立標準技術研究所（National Institute of Standards and Technology：NIST）は，これまでに，5 件の標準化候補暗号方式（鍵交換 2 件，デジタル署名 3 件）を選定し，そのうちの 3 件（鍵交換 1 件，デジタル署名 2 件）に関しては，標準化が完了している [22]．それぞれの暗号技術を記載した標準化文書は，FIPS 203 [19], FIPS 204 [20], FIPS 205 [21] として 2024 年 8 月に公開された．残り 2 方式に関しても，標準化作業が進んでおり，近いうちに公開されることが予定されている．さらに，デジタル署名技術の追加公募も行わ

[*9] 素因数分解問題や離散対数問題の求解に関する量子アルゴリズムを含む，これまでに知られた量子アルゴリズムは，量子アルゴリズムの研究者であるステファン・ジョーダン博士が運営する Quantum Algorithm Zoo のサイト（`https://quantumalgorithmzoo.org/`）にまとめられており，誰でもみることができる．

れており，現在選定作業が進行中である．また，国内では，CRYPTREC[*10]が耐量子計算機暗号の電子政府推奨暗号リストへの掲載に向けた安全性評価・実装性能評価を開始している [6].

本稿では，上記 (1) に焦点を当て，素因数分解問題と離散対数問題の求解に関する量子アルゴリズムの解説とその研究の進展について整理する．特に，素因数分解問題を求解する量子アルゴリズムに必要な量子ビット数は，1994 年のショアによるアルゴリズムを 1 としたとき，2017 年には 1/4 まで低下しており，今後も更なる効率化が見込まれている．こうした状況も踏まえると，情報セキュリティ業務に従事するうえでは，暗号技術の安全性評価について理論的な理解を持ち合わせておくことが望ましい．もっとも，隠れ部分群問題や量子アルゴリズムの理解には，純粋数学や量子力学に関する高度な専門知識が必要であり，情報セキュリティの専門家であっても理解が難しいのが実情である．そこで，本稿では，量子コンピュータに対する従来暗号技術の安全性評価に関する知識の底上げを目的とし，情報技術に関する一般的な知識で理解できるよう解説することとしたい.

本稿の構成は以下のとおりである．まず，2 章では，素因数分解問題と離散対数問題について，それぞれ，古典コンピュータで用いられるアルゴリズム（2.1 章）と，量子コンピュータで用いられる量子アルゴリズム（2.2 章）の概要を紹介する．

3 章と 4 章では，隠れ部分群問題と呼ばれる数学的問題を求解する量子アルゴリズムを通して，素因数分解問題と離散対数問題が求解できることを示す．3 章では，(1) 隠れ部分群問題，(2) 1 変数周期関数の周期を求める問題，(3) 素因数分解問題の 3 つの数学的問題の関係に着目する．これらの問題については，量子アルゴリズムに関する研究の成果として，包含関係にあることがわかっている．(1) 隠れ部分群問題は，(2) 1 変数周期関数の周期を求める問題を包含し，(2) 1 変数周期関数の周期を求める問題は，(3) 素因数分解問題を包含する関係にある (すなわち，(1) $\supset$ (2) $\supset$ (3))．このため， (1) の問題を求解する量子アルゴリズムを利用して (2) の問題を求解できるほか，(2) の問題を求解する量子アルゴリズムを利用して (3) の問題を求解できる関係となっている．このうち，3 章では，(1) の問題と (2) の問題の包含関係について説明を行うとともに（3.1 章），(2) の問題を求解する量子アルゴリズムを解説したうえで（3.2 章），その拡張により (3) の問題が求解できることを説明する（3.3 章）.

4 章では，(1) 隠れ部分群問題，(4) 2 変数全射準同型写像の核を求める問題，(5) 離散対数問題の 3 つの数学的問題の関係に着目する．これらの問題についても，量子アルゴリ

[*10] 電子政府推奨暗号の安全性を評価・監視し，暗号技術の適切な実装法・運用法を調査・検討するプロジェクト.

ズムに関する研究の成果として，(1) 隠れ部分群問題は，(4) 2 変数全射準同型写像の核を求める問題を包含し，(4) 2 変数全射準同型写像の核を求める問題は，(5) 離散対数問題を包含する関係にあることがわかっており，(4) の問題を求解する量子アルゴリズムを利用して (5) の問題を求解することができる (すなわち，(1) $\supset$ (4) $\supset$ (5))．このうち，4 章では，(1) の問題と (4) の問題の包含関係について説明を行うとともに（4.1 章），(4) の問題を求解する量子アルゴリズムの概要を解説したうえで（4.2 章），(5) の問題を求解する量子アルゴリズムの前提となる考え方を説明する（4.4 章）．なお，(5) の問題の解は，(4) の問題の解から得られるため，(4) を求解するアルゴリズムの理解が重要となる．このアルゴリズムは，問題の設定の仕方によって複数の解き方が考えられることから，4.2 章では大枠のみを説明することとし，具体的なアルゴリズムの詳細については，別途 5 章で説明を行う．さらに，(4) の問題について，変数の数を増やした問題 (n 変数全射準同型写像の核を求める問題) に拡張する場合も取り上げる (4.3 章)．この拡張した問題を求解する量子アルゴリズムを (5) の問題に適用すれば，複数の同問題を同時に求解できる (5.5 章)．

5 章では，(4) の問題を求解するアルゴリズムをベースに，(5) の問題を求解する量子アルゴリズムについて解説を行う（5.1〜5.3 章）．その際，問題の設定の仕方によっては，離散対数問題の解を使って素因数分解問題を求解することができることを示す（5.4 章）．(4) の問題を多変数とすることで，複数の離散対数問題を同時に求解する量子アルゴリズムについては，5.5 章で説明する

最後に，6 章で本稿のまとめを行う．

第 2 章

素因数分解問題と離散対数問題の求解

2.1　古典コンピュータを前提とした安全性評価

現在広く利用されている RSA 暗号や ECDSA 等は，古典コンピュータを前提とした計算量的安全性を有する暗号技術である．こうした従来暗号技術については，多くの暗号学者によってその安全性評価が行われてきており，現時点において，古典コンピュータで素因数分解問題や離散対数問題を求解する効率的なアルゴリズムが見つかっていないという意味で安全であるといわれている．このため，従来暗号技術の安全性評価では，素因数分解や離散対数問題の求解がどの程度難しいかが問題となる．現在知られている最も効率的なアルゴリズムの計算量は，問題の入力長[*11]に対して準指数関数[*12]で表されることから，問題の入力長を適切に設定することで現実的な求解を困難とすることができる．現時点では，2048 ビットの合成数であれば，スーパーコンピュータであっても現実的な時間で素因数分解することが困難と評価されていることから，RSA 暗号の公開鍵には 2048 ビットの合成数が利用されている [7]．このように，安全な公開鍵のサイズは，最新の技術進展を踏まえた計算量評価に基づいている．

古典コンピュータの性能は技術進展に伴い向上していくが，より効率的なアルゴリズムが発見されない限りにおいては，公開鍵長を伸長しさえすれば，暗号技術の安全性を維持することが可能であった．実際，RSA 暗号の鍵長は，768 ビット，1024 ビット，2048 ビッ

[*11] 素因数分解問題であれば分解対象の合成数の，離散対数問題であれば法のビット長をいう．

[*12] 問題の入力長 n に対して，n の指数関数と多項式の間に属する関数．求解に準指数関数時間かかる問題は計算が困難である（現実的な時間で計算できない）とみなされる．

トと伸長してきており，2031 年以降は 3072 ビットとすることが推奨されている [8][*13].

2.2 量子コンピュータを前提とした安全性評価

1994 年，素因数分解問題と離散対数問題は，量子コンピュータを利用することで，現実的な時間（多項式時間[*14]）で求解可能であることがショアによって示された [24].

素因数分解問題について，ショアが示した量子アルゴリズムは，合成数 N と N と互いに素な自然数 a に対して，$a^r \bmod N = 1$ となる自然数 r を求め，r から N の素因数を求めるものである．量子コンピュータでは，大量の x に対する $a^x \bmod N$ の計算結果を重ね合わせの状態として保持できることから，計算結果の絞込みによって $a^r \bmod N = 1$ となる r を効率的に見つけることが可能である．詳細は 3 章で解説する．

離散対数問題については，量子アルゴリズムに関する研究の進展を受けて，全射準同型写像の核を求める問題の量子アルゴリズムを使って解けることがわかっている．いま，$h = g^s \bmod p$（p は素数）を満たす g と h に関して，$f(x, y) = g^x h^y \bmod p$ としたとき，$f(x, y) = 1$ を満たす (x, y) の集合は，$f(x, y)$ の核と呼ばれる．このとき，$(s, -1)$ は $f(x, y)$ の核の基底であることから，$f(x, y)$ の核を求める問題が解ければ，離散対数問題（$h \equiv g^s \pmod p$）の解 s を求めることができるというものである．上記の周期関数の周期を求める問題と同様，量子コンピュータでは，大量の x と y に対する $f(x, y)$ の計算結果を重ね合わせの状態として保持できることから，計算結果の絞込みによって $f(x, y) = 1$ となる x と y を効率的に求めることができる．全射準同型写像の核を求める量子アルゴリズムについては，4 章で解説する．

さらに，離散対数問題を，群の位数[*15]が既知の場合と未知の場合，および，解の大きさに制限がある場合と制限がない場合の 4 つにわけたとき，位数が未知で解の大きさに制限がある離散対数問題については，2016 年，エケラによって効率的な量子アルゴリズムが示された [13]．エケラによる量子アルゴリズムでは，解の大きさに上限をもたせることで，計算に必要な量子ビットを少なくできるという特徴があるほか，量子コンピュータを用いて求めた値から解を求める計算方法を変更したことで効率化が図られた．さらに，2017 年にはエケラらによって，より少ない量子ビットで離散対数問題を求解する量子アルゴリズムが提案されたほか，離散対数問題の解を用いて素因数分解できることが示

[*13] 電子政府システムにおいては，鍵長を 2048 ビットとする RSA 暗号を 2031 年以降も使用することは原則認められておらず，3072 ビットとする RSA 暗号に移行しなければならない [8].

[*14] 問題の入力サイズ n に対して，n の多項式（例えば，n^3）で表される計算時間.

[*15] 集合の要素数.

された [14]．具体的には，$h \equiv g^s \pmod N$（N は合成数）について，s を求めることができれば，そこから得られる情報から合成数 N を素因数分解できるというのが，エケラらによる結果である．これにより，位数が未知の離散対数問題は，周期関数 $f(x) = g^x \pmod N$ の周期を求める問題とみることもできるため，準同型写像の核を求める問題でもあり，かつ，周期関数の周期を求める問題でもあると整理することができる．これらの問題の関係をまとめると図 2 のように整理できる．なお，非可換群[*16]の隠れ部分群問題については，現時点で多項式時間で計算可能な量子アルゴリズムは知られていない．そのため，耐量子計算機暗号では，非可換群のうち，二面体群[*17]に対する隠れ部分群問題に含まれる格子問題[*18]が利用されている．

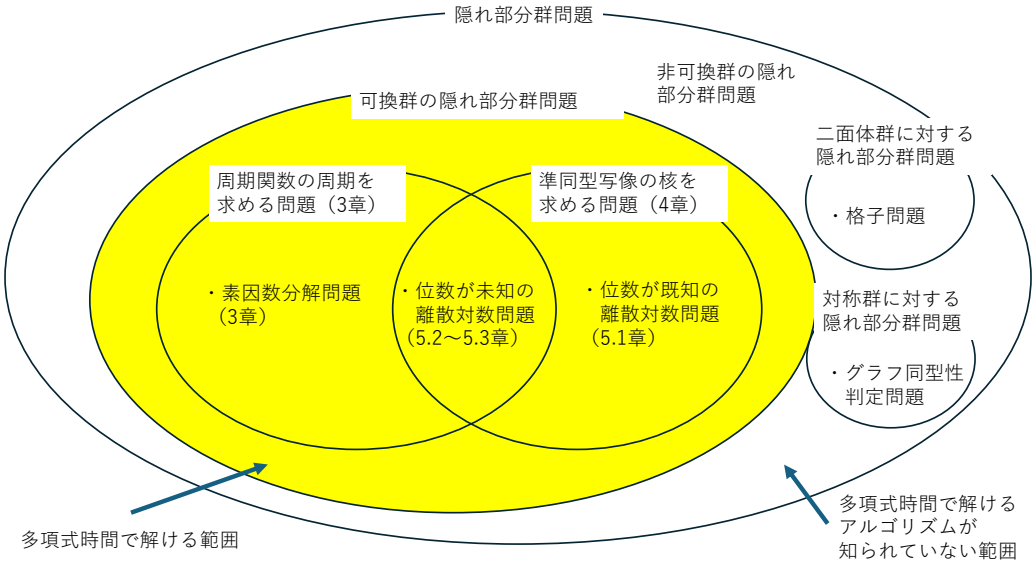

図 2　隠れ部分群問題と素因数分解問題・離散対数問題との関係

[*16] 可換群でない群．

[*17] 正多角形を自分自身に移す合同変換（形を変えずに位置を移動させる変換）の集合

[*18] 格子（規則的な網目の交点の集合）に関する問題であり，その代表的な問題には最短ベクトル問題がある．最短ベクトル問題とは，基底ベクトルから生成される格子点のうちで原点に最も近いベクトルを見つける問題．基底ベクトルとは，2 次元であれば空間上のあらゆるベクトルを $\vec{v} = a\vec{x} + b\vec{y}$ のかたちで表すことを可能にするベクトル $\vec{x}, \vec{y}$ の組み合わせをいう．

第 3 章

素因数分解問題を求解する量子アルゴリズム

本章では，素因数分解問題を求解する量子アルゴリズムについて解説する．素因数分解問題は，隠れ部分群問題に属する 1 変数周期関数の周期を求める量子アルゴリズムを用いて求解できる．そこで，まず，1 変数周期関数の周期を求める問題を紹介し，同問題が隠れ部分群に含まれることを確認する．そのうえで，1 変数周期関数の周期を求める問題を経由することで素因数分解ができることを示す．本稿では，$\mathbb{Z}$ は整数の集合とする．

まず，隠れ部分群問題を簡単に説明する．

定義 1 (隠れ部分群問題)**.** 有限生成群[*19]H と有限集合 X に対して，H から X へのブラックボックス関数[*20]f が与えられており，任意の $x, y \in H$ に対して，

$$f(x) = f(y) \iff x - y \in K$$

が成り立つとする．ここで，K は H の未知の部分群である．このとき，K を定めよ．

3.1 1 変数周期関数の周期を求める問題と隠れ部分群問題

1 変数周期関数の周期を求める問題は以下で定義される．

[*19] n を自然数として $\mathbb{Z}^n$，もしくは q を自然数として，$\mathbb{Z}_q^n$ と考えてよい．これらは可換群であり，特に断りのない限り，可換群のみを扱う．

[*20] 外部からは内部構造や動作原理は不明であり，入力に対して，出力値しかわからない関数である．

問題 1 (1 変数周期関数の周期を求める問題)**.** S を有限集合とする．関数 $f(x)$ を，$\mathbb{Z}$ から S への全射とし，周期が r であるとする．つまり，ある自然数 r が存在して，すべての $x \in \mathbb{Z}$ に対して，$f(x+r) = f(x)$ が成り立つとする．このとき，最小の r を求めよ．

この問題を求解する量子アルゴリズムを理解するうえで必要となる周期関数に関する情報を以下に整理する．なお，数学的準備については，付録 A を参考にされたい．

- f は全射であるので，$f(\mathbb{Z}) \coloneqq \{f(x) \mid x \in \mathbb{Z}\} = S$ であり，$|f(\mathbb{Z})| = |S| = r$ である．
- $K = r\mathbb{Z} = \{, \ldots, 0, r, 2r, 3r, \ldots\}$ とする（このとき，K は r により生成される）．すべての $k \in K$ に対して，$f(k) = f(0)$ である．
- $0 \leq t < r$ に対して，集合 K_t を

$$K_t := t + K = \{t + k \mid k \in K\} = \{\ldots, t - r, t, t + r, t + 2r, \ldots\}$$

  とする．このとき，すべての $k \in K_t$ に対して，$f(k) = f(t)$ が成り立つ．
- $t \neq s$ のとき，$f(K_t) \neq f(K_s)$ である．
- $\bigcup_{t=0}^{r-1} K_t = \mathbb{Z}$ である．

問題 1 は，$K = r\mathbb{Z}$ と考えることにより，隠れ部分群問題に含まれる．この事実を確認する．もし，$f(x) = f(y)$ であれば，常に，ある整数 k が存在して，$x = y + kr$ と書くことができ，$x - y = kr \in r\mathbb{Z}$ となる．逆に，$x - y \in r\mathbb{Z}$ であれば，ある整数 k が存在して，$x = y + kr$ と書くことができ，$f(x) = f(y + kr) = f(y)$ が成り立つ．以上より，問題 1 は，$\mathbb{Z}$ に対する $K = r\mathbb{Z}$ とした隠れ部分群問題となる．

N を合成数とし，$a \in \mathbb{Z}_N^*$ とする．関数 $f(x)$ を，

$$f(x) = a^x \bmod N \tag{3.1}$$

とする．r を a の N を法とした[*21]（未知の）位数とするとき[*22]，$f(x+r) = a^{x+r} \bmod N = a^x a^r \bmod N = a^x \bmod N = f(x)$ であるため，f は周期 r の周期関数である．

例 1 ($N = 15$ の例)**.** $N = 15, a = 7$ とする．$f(x) = 7^x \bmod 15$ を考えるとき，$f(0) = 7^0 \bmod 15 = 1, f(4) = 7^4 \bmod 15 = 1$ が成り立つほか，$f(1) = 7^1 \bmod 15 = 7 \neq 1, f(2) = 7^2 \bmod 15 = 4 \neq 1, f(3) = 7^3 \bmod 15 = 13 \neq 1$ であるため，周期は

[*21] ここで，「法とする」とは，N で除算をした余りを考えることを意味する．

[*22] つまり，r は $a^r \bmod N = 1$ を満たす最小の自然数である．

$r = 4$ である．このとき，$f(x+4) = 7^{x+4} \bmod 15 = 7^x 7^4 \bmod 15 = 7^x \bmod 15 = f(x)$ も成り立つことから，f は周期 4 の周期関数である．

3.2　1 変数周期関数の周期を求める問題の量子アルゴリズム

1 変数周期関数の周期 r を求める量子アルゴリズムは，以下の (1), (2) で構成される．なお，m を事前に定めた自然数とする．

(1)　l/r の m ビット近似を求める．ここで，l は 0 から $r-1$ の範囲のランダムな整数である．

(2)　l/r の m ビット近似値から，連分数展開[*23]を用いることにより，r（および l）を求める．

また，関数 f は $\mathbb{Z}$ 上で定義されているが，量子コンピュータ上での実装を想定し，$0 \leq x \leq 2^m - 1$ で打ち切ることにする．

以下の l/r の m ビット近似を求める量子アルゴリズムでは 2 つの量子レジスタ[*24]を利用する．第 1 量子レジスタは，m 個の量子ビットにより構成され，第 2 量子レジスタは，すべての x に対して，$f(x)$ の値を保持するのに十分な量子ビット数により構成される．ここでは，n 量子ビットであるとする．

ステップ 1：　$m+n$ 量子ビットからなる初期状態 $\underbrace{|0\rangle \cdots |0\rangle}_{m\text{ 個}} \underbrace{|0\rangle \cdots |0\rangle}_{n\text{ 個}}$ を用意する．

ステップ 2：　第 1 量子レジスタに対してアダマールゲートを作用させる．

$$\underbrace{|0\rangle \cdots |0\rangle}_{m\text{ 個}} |0\rangle \to \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |0\rangle$$

ステップ 3：　第 2 量子レジスタに，$f(x)$ を書き込む．

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |0\rangle \to \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |f(x)\rangle$$

ステップ 4：　第 1 量子レジスタに逆量子フーリエ変換を作用させる．

[*23] この計算は古典コンピュータ上で実行可能である．

[*24] 量子レジスタは，量子ビットの値を格納する領域である．古典コンピュータにおけるレジスタの量子版に対応する．

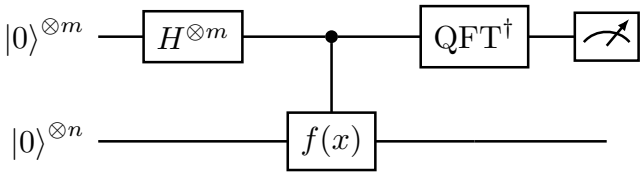

図 3　回路図（1 変数周期関数の周期を求める回路）

ステップ 5：　第 1 量子レジスタを測定する．ここで，測定値は $2^m l/r$ の小数点以下四捨五入した値となる．この値を 2^m で割ることにより，l/r の m ビット近似を得る．

回路図を図 3 に記す．なお，$|0\rangle^{\otimes m}$ とは，$\underbrace{|0\rangle \cdots |0\rangle}_{m\text{ 個}}$ を示し，$H^{\otimes m}$ は，m 個の量子ビットそれぞれにアダマールゲートを作用させることを示す．また，$\mathrm{QFT}^\dagger$ は，逆量子フーリエ変換を示す．

ステップ 5 での測定により，確かに l/r の m ビット近似が得られることを確認する．簡単のため，$2^m/r$ が自然数 u であるとする．ステップ 3 終了時の量子状態は，f の周期が r であることを考慮すると，

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle\,|f(x)\rangle = \frac{1}{\sqrt{2^m}} \sum_{g=0}^{r-1} \sum_{t=0}^{u-1} |g+rt\rangle\,|f(g)\rangle$$

と書くことができる．ステップ 4 以降では，第 2 レジスタは計算結果に影響を与えないので，第 2 レジスタ部（$|f(g)\rangle$）を無視することにする．第 1 レジスタを逆量子フーリエ変換すると，

$$\frac{1}{\sqrt{u}} \sum_{t=0}^{u-1} |g+rt\rangle \xrightarrow{\mathrm{QFT}^\dagger} \frac{1}{\sqrt{u2^m}} \sum_{k=0}^{2^m-1} \sum_{t=0}^{u-1} \exp\left(-\frac{2\pi \mathrm{i}(g+rt)k}{2^m}\right) |k\rangle$$

となる．

第 1 レジスタの測定により k が得られる確率は，

$$\left| \frac{1}{\sqrt{u2^m}} \sum_{t=0}^{u-1} \exp\left(-\frac{2\pi \mathrm{i}(g+rt)k}{2^m}\right) \right|^2 = \frac{1}{u2^m} \left| \sum_{t=0}^{u-1} \exp\left(-\frac{2\pi \mathrm{i}kt}{u}\right) \right|^2$$

となる．

k が u の倍数のときを考える．$k = ul$ とおくと，

$$\frac{1}{u2^m} \left| \sum_{t=0}^{u-1} \exp\left(-\frac{2\pi \mathrm{i}kt}{u}\right) \right|^2 = \frac{1}{u2^m} \left| \sum_{t=0}^{u-1} \exp(-2\pi \mathrm{i}lt) \right|^2 = \frac{u^2}{u2^m} = \frac{u}{2^m} = \frac{1}{r}$$

となり，$k = ul$ が得られる確率は $1/r$ となる．

k が u の倍数ではないときを考える．$j = 1, \ldots, u-1$ に対して，$k = ul + j$ とおくと，

$$\frac{1}{u2^m}\left|\sum_{t=0}^{u-1}\exp\left(-\frac{2\pi \mathrm{i}kt}{u}\right)\right|^2 = \frac{1}{u2^m}\left|\sum_{t=0}^{u-1}\exp\left(-\frac{2\pi \mathrm{i}(ul+j)t}{u}\right)\right|^2$$
$$= \frac{1}{u2^m}\left|\sum_{t=0}^{u-1}\exp\left(-\frac{2\pi \mathrm{i}jt}{u}\right)\right|^2 = 0$$

となり，$j = 1, \ldots, u-1$ に対して，$k = ul + j$ が得られる確率は 0 になる．

以上より，第 1 量子レジスタを測定することにより，第 1 レジスタに格納されている値，つまり，$ul = 2^m l/r$ が得られることになる．これを 2^m で割ることにより，l/r が得られる．

以上の説明では，$2^m/r$ が自然数であるという特殊なケースを考えていたが，自然数とならない場合でも同様の議論が可能であり，実際，このアルゴリズムを動作させることにより，$2^m l/r$ を小数点以下四捨五入した値を高い確率で得ることができる．最終的には，l/r を連分数展開（付録 A.3 を参照）することにより，周期 r（と l）を求めることができる．

3.3 素因数分解問題を求解する量子アルゴリズム

ショアは，3.2 節で解説した周期関数の周期を求める量子アルゴリズムを発展させることで素因数分解問題を求解する方法を示した．具体的には，素因数分解の対象である合成数 N に対して，周期関数 $a^r \mod N$ の周期 r を求めることができれば，r を使って N を素因数分解できることを示したものである．

ここでは，ショアにより提案された素因数分解問題を求解する量子アルゴリズム [24] を説明する．ショアのアルゴリズムは以下の (1) と (2) で与えられる．

(1) 素因数分解のターゲットである合成数 N，N と互いに素な自然数 a に対して，

$$a^r \bmod N = 1$$

となる自然数 r を求める．

(2) r を用いることにより，高い確率で N の素因数分解を行う．

いま，関数 $f_a(x)$ を $f_a(x) = a^x \bmod N$ とする．この関数は，位数 r に対して，$f_a(x+r) = f_a(x)$ となるため，$f_a(x)$ は周期 r の関数である．そのため，3.2 章で述べた

1 変数関数の周期を求める量子アルゴリズムを用いることにより，r を求めることが可能である[*25].

ここで, r を偶数であるとする．議論を簡単にするため, 素因数分解をする合成数 N は, 同サイズの異なる素数 p, q の積とする．$a^r \bmod N = 1$ を変形すると，$(a^{r/2})^2 \bmod N = 1$ が得られるため，複号任意で $a^{r/2} \bmod p = \pm 1,\ a^{r/2} \bmod q = \pm 1$ が成り立つ．これにより，4 個の場合分け

$$(a^{r/2} \bmod p,\ a^{r/2} \bmod q) = (+1,+1), (+1,-1), (-1,+1), (-1,-1)$$

が生じる．この 4 個のうち，以下の 2 つ

- $a^{r/2} \bmod p = +1$ かつ $a^{r/2} \bmod q = -1$
- $a^{r/2} \bmod p = -1$ かつ $a^{r/2} \bmod q = +1$

のとき，最大公約数 $\gcd(a^{r/2} - 1, N)$ を計算することにより，素因数分解が可能である．具体的に，前者の場合は，$\gcd(a^{r/2} - 1, N) = p$，後者の場合は，$\gcd(a^{r/2} - 1, N) = q$ が成り立つことがわかる．

なお，3.2 章で述べたアルゴリズムのステップ 3 では，第 2 量子レジスタに $f(x)$ を書き込む必要がある．素因数分解で用いる周期関数 $f_a(x) = a^x \bmod N$ においては，第 2 レジスタへの $f_a(x)$ の書き込みは，多項式時間で実現可能ではあるものの，量子アルゴリズムのほかの部分（ステップ 2 でのアダマール変換およびステップ 4 での逆量子フーリエ変換など）と比較すると，多くの計算時間が必要である．そのため，現在も効率化に関する研究が行われている．

必要となる量子ビット数について説明する．連分数展開により，高い確率で正しい r を求めるためには，第 1 レジスタとして，$2 \log N$ ビット必要である[*26]．いま，N の素因数 p, q をそれぞれ n ビットとすると，$4n$ 量子ビットの量子レジスタが必要となる．ただし，これ以外に $a^x \bmod N$ を計算する第 2 レジスタが必要であることに注意されたい．

また，前述の $a^r \bmod N = 1$ となる自然数 r を求める問題は，合成数（$N = pq$）を法とした位数（$p-1$ と $q-1$ の最小公倍数）が未知の離散対数問題とみなすこともできる．このアイデアを発展させて，離散対数問題の解を使用して素因数分解問題を求解する量子アルゴリズムについては，5.4 章で説明する．

[*25] 周期関数の周期を求める問題ではなく，位相推定アルゴリズム問題を経由して，素因数分解を行うアプローチも知られている．詳しくは， [3] などを参照されたい．アプローチは異なるものの，アルゴリズムは同一である．

[*26] 詳細は，A.3 を参照されたい

第 4 章

離散対数問題と全射準同型写像の核を求める問題

本章では，5 章で解説する離散対数問題を求解する量子アルゴリズムの理解に必要となる，全射準同型写像の核を求める問題について解説する．全射準同型写像の核を求める問題は，隠れ部分群問題の部分問題であることを示したうえで，離散対数問題が同問題に帰着できることを確認する．さらに，一般の n 変数の全射準同型写像にまで拡張する．なお，実際の離散対数問題を求解するアルゴリズムは，5 章で説明する．

4.1 2 変数全射準同型写像の核を求める問題と隠れ部分群問題

隠れ部分群問題における写像 f に制限を加えることで，以下の 2 変数全射準同型写像（全射準同型写像については．付録 A.2 を参照）の核を求める問題が定義できる．なお，核とは準同型写像 f によって単位元に写る元の集合である．

問題 2. G を有限可換群とし，関数 $f(x_1, x_2)$ を $\mathbb{Z}^2$ から G への全射群準同型写像とする．このとき，写像 f の核 $\ker(f)(:= \{(x, y) \in \mathbb{Z}^2 \mid f(x, y) = 1\})$ を定めよ（1 は G の単位元とする）．

問題 2 が隠れ部分群問題に含まれることを示す．いま，$f(x_1, y_1) = f(x_2, y_2)$ が成り立つとする．このとき，$f(x_1, y_1)/f(x_2, y_2) = 1$ であるほか，f は準同型写像であることから，$f(x_1 - x_2, y_1 - y_2) = 1$ が成り立つ．そのため，$(x_1 - x_2, y_1 - y_2) = (x_1, y_1) - (x_2, y_2) \in \ker(f)$ である．同様に，$(x_1, y_1) - (x_2, y_2) = (x_1 - x_2, y_1 - y_2) \in \ker(f)$ であれば，

$f(x_1, y_1) = f(x_2, y_2)$ が成り立つ．そのため，問題 2 は，$\mathbb{Z}^2$ における未知の部分群 $\ker(f)$ を求める隠れ部分群問題である．

以下では，次節での離散対数問題の求解に使用するため，f の形を制限した場合の $\ker(f)$ について説明する．位数 q の有限可換群 G の要素 $g_1, g_2 \in G$ に対して，f を以下の $\mathbb{Z}^2$ から G への上への写像

$$f(x, y) = g_1^x g_2^y$$

とする[*27]．f は，

$$f(x_1, y_1)f(x_2, y_2) = g_1^{x_1} g_2^{y_1} g_1^{x_2} g_2^{y_2} = g_1^{x_1+x_2} g_2^{y_1+y_2} = f(x_1 + x_2, y_1 + y_2)$$

であるため，$\mathbb{Z}^2$ から G への全射準同型写像となる．このとき，写像 f に対する $\ker(f)$ は，2 次元の格子[*28]となり，2 個の基底により生成される．具体的には，$(q, 0)$ およびあるベクトル (s_1, s_2) により生成される．つまり，$\ker(f)$ のすべての要素は，ある整数 a, b に対して，$a(q, 0) + b(s_1, s_2)$ と記述することができる[*29]．

定義域を $\mathbb{Z}_q^2$ とした上で，同じ写像 $f(x, y) = g_1^x g_2^y$ を考えるとき，f は同様に準同型写像となる．このとき，$\ker(f)$ は 1 次元の格子となる．つまり，あるベクトル $(s_1, s_2) \in \mathbb{Z}_q^2$ が存在して，$\ker(f) = \{l(s_1, s_2) \mid l \in \mathbb{Z}_q\}$ と書くことができる．

4.2　2 変数全射準同型写像の核を求める量子アルゴリズム

本節では，全射準同型写像の核を求めるアルゴリズムの概略を示す．この問題の求解には，量子コンピュータ（ステップ 1 の処理）と古典コンピュータ（ステップ 2 の処理）の両方を使用する．以下では，量子コンピュータによる処理を量子計算，古典コンピュータによる処理を古典計算と呼ぶ．

ただし，x を非負の実数として，$x \bmod 1$ は，x の小数部分とする[*30]．また，m を自然数とする．

ステップ 1（量子計算）　5 章で述べる量子アルゴリズムにより，

$$(ls_1/q \bmod 1, ls_2/q \bmod 1)$$

[*27] 式 (3.1) の自明な拡張である．

[*28] $m \geq n$ を満たす自然数に対し，R^m の n 個のベクトル $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ による整数係数結合全体の集合 $\{\sum_{i=1}^n x_i \boldsymbol{b}_i \mid x_1, \ldots, x_n \in \mathbb{Z}\}$ を R^m の次元 n の格子と呼ぶ．$\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ を基底と呼ぶ．

[*29] $(0, q)$ も，この形で記述することが可能である．

[*30] 例えば，$1.3 \bmod 1 = 0.3$ である．

の小数点以下 2 進で m ビットの近似を得る．ここで，l は 0 から $q-1$ の範囲のランダムな自然数である．$s_1 = 1$ としても，一般性は失わないため，以降．$s_1 = 1$ とし，s_2 ではなく，単に s と書くことにする．つまり，$(l/q, ls/q \bmod 1)$ が得られるとする[*31]．

ステップ 2 （古典計算） 測定値から，古典コンピュータを用いて，s，q，l を求める．

ステップ 2 において，測定値（以降，(α_1, α_2) とする）から，s，q，l を求める方法について，場合分けをして説明する．

1. q が既知のときを考える．$l/q \approx \alpha_1$ であるため，$l \approx \alpha_1 q$ が成り立つ．これより，$\alpha_1 q$ を四捨五入することにより，l を得る．また，$s \approx \alpha_2 q/l$ であるため，この値を四捨五入することにより，s を得る．
2. q が未知のときを考える．α_1 を連分数展開することにより，q と l を得る[*32]．さらに，$s \approx \alpha_2 q/l$ であることから，この値を四捨五入することにより，s を得る．

q が既知であるか未知であるかによって，ステップ 1 の計算に必要な量子ビット数が異なることに注意されたい．q が既知のときは，未知変数は，l, s の 2 個あり，おのおの $\log q$ ビットであるため，少なくとも全体で $2\log q$ 量子ビットが必要である[*33]（さらに，最終量子レジスタ分および計算に必要な補助ビットが必要である）．q が未知のときは，さらに，q も求める必要があるため，全体で $3\log q$ 量子ビットが必要である（同様に，最終量子レジスタ分および計算に必要な補助ビットが追加で必要である）．詳しくは，5 章で説明する．

4.3 n 変数全射準同型写像の核を求める問題

前節では，2 変数全射準同型写像の核を求めることで離散対数問題が求解できることを説明した．本節では，この全射準同型写像の変数を増やすことで複数の離散対数問題を同時に求解できることを示す．

n を自然数として，n 変数全射準同型写像の核を求める問題を以下で定義する．

問題 3. G を有限可換群とする．関数 $f(x_1, \ldots, x_n)$ は，$\mathbb{Z}^n$ から G への全射群準同型写像とする．このとき，$\ker(f) (:= \{(x'_1, x'_2, \ldots, x'_n) \in \mathbb{Z}^n \mid f(x'_1, x'_2, \ldots, x'_n) = 1\})$ を定め

[*31] $l < q$ であるため，$l/q < 1$ が成り立つ．そのため，mod1 は省略可能である．
[*32] この計算は，1 変数周期関数における周期を計算するステップと同一である
[*33] 実際は，解の精度を保証するため追加のビットが必要である．

よ（ここで，1 は G の単位元である）．

2 変数の場合と同様に，n 変数の場合も，

$$\begin{aligned} &f(x'_1,\ldots,x'_n) = f(x''_1,\ldots,x''_n) \\ &\iff (x'_1,\ldots,x'_n) - (x''_1,\ldots,x''_n) = (x'_1 - x''_1,\ldots,x'_n - x''_n) \in \ker(f) \end{aligned}$$

が成り立つため，n 変数全射準同型写像の $\ker(f)$ を求める問題は隠れ部分群問題に含まれる．

以下，f の形および定義域を制限し，位数 q の有限可換群 G を考える．$g_1, g_2, \ldots, g_n \in G$ に対して，写像 f を $\mathbb{Z}_q^n$ から G への上への写像として，

$$f(x_1, x_2, \ldots, x_n) = g_1^{x_1} g_2^{x_2} \cdots g_n^{x_n}$$

で定義する．このとき，f は $\mathbb{Z}_q^n$ から G への全射準同型写像となる．g を位数 q の元として，各 g_i は，$g_i = g^{\alpha_i}$ と記述できるとする．このとき，

$$f(x_1,\ldots,x_n) = g^{\alpha_1 x_1 + \cdots + \alpha_n x_n}$$

となる．これより，$\alpha_1 x_1 + \cdots + \alpha_n x_n \equiv 0 \pmod{q}$ のとき，$f(x_1,\ldots,x_n) = 1$ となる．逆に，$f(x_1,\ldots,x_n) = 1$ となるのは，$\alpha_1 x_1 + \cdots + \alpha_n x_n \equiv 0 \pmod{q}$ のときのみである．これより，

$$\ker(f) = \{(x_1,\ldots,x_n) \in \mathbb{Z}_q^n \mid \alpha_1 x_1 + \cdots + \alpha_n x_n \equiv 0 \pmod{q}\}$$

となる．このとき，写像 f に対する $\ker(f)$ は，$n-1$ 次元の格子をなし，$n-1$ 個の基底により生成される．具体的には，$n-1$ 個の基底ベクトル $\vec{s}_1,\ldots,\vec{s}_{n-1} \in \mathbb{Z}_q^n$ に対して，

$$\ker(f) = \{l_1\vec{s}_1 + \cdots + l_{n-1}\vec{s}_{n-1} \mid l_1,\ldots,l_{n-1} \in \mathbb{Z}_q\}$$

となる．ここで，ベクトル $\vec{\alpha} = (\alpha_1,\ldots,\alpha_n)$ とすると，すべての基底ベクトルに対して，$\vec{\alpha}\cdot\vec{s}_i = 0$ が成り立つ[*34]ため，$\vec{\alpha}$ と $\vec{s}_i = 0$ は直交する．このため，すべての $\vec{v} \in \ker(f)$ に対して，$\vec{\alpha}\cdot\vec{v} = 0$ である．

次に，核 $\ker(f)$ を求める量子アルゴリズムを説明する．この量子アルゴリズムでは，まず，$(\alpha_1,\ldots,\alpha_n)$ を求める[*35]．これは，5 章で述べる量子アルゴリズムを n 変数版に拡張することにより行われる．ついで，$\ker(f)$ に対する基底を求める．$\ker(f)$ の要素を複数回ランダムに選び，最終的に線形独立な $n-1$ 個のベクトルを選ぶ．この $n-1$ 本のベクトルが，基底 $\vec{s}_1,\ldots,\vec{s}_{n-1}$ となる．

[*34] $\vec{a}\cdot\vec{b}$ を，2 つのベクトル $\vec{a},\vec{b}$ に対する内積とする．

[*35] ここで，$\alpha_1 = 1$ として十分である．

4.4 2 次元全射準同型写像の核を求める問題と離散対数問題との関係

ここでは，2 次元全射準同型写像 $f(x,y) = g^x h^y$ における核の基底を求めることによって，離散対数問題 $h = g^s$ の解 s を求めることができることを示す．

G を素数位数 q を持つ可換群とし，群 G 上での離散対数問題（$g, h \in G$ が与えられたときに，$h = g^s$ となる s を求める問題）を考える．関数 f を，$\mathbb{Z}_q^2$ から G への写像とし，

$$f(x,y) = g^x h^y$$

とする．このとき，以下が成り立つ．

(1) $l = 0, 1, \ldots, q-1$ として，$\ker(f) = \{l(s,-1) | l \in \mathbb{Z}_q\} = \{(ls, -l \bmod q) \mid l \in \mathbb{Z}_q\}$ となる．実際，$f(ls, -l \mod q) = g^{ls} h^{-l} = g^{ls} g^{-ls} = g^0 = 1$ となる．また，$\ker(f)$ は 1 次元線形空間となるので，基底は 1 つのみである．これより，$(s,-1)$ は $\ker(f)$ の基底である

(2) $0 \leq t < q-1$ に対して，集合 K_t を

$$K_t := \{(k_1, k_2) | k_1 + sk_2 \bmod q = t\}$$

とする．すべての $(k_1, k_2) \in K_t$ に対して，$f(k_1, k_2) = g^t$ である．

(3) $t \neq s$ のとき，$f(K_t) \neq f(K_s)$ である[*36]．

関数 f が与えられたときに，もし，$\ker(f)$ の基底 $(s,-1)$ を求めることができれば，第 1 要素は離散対数問題の解 s であることから，離散対数問題の解 s を求めることが可能である．

$q = 3$ の場合を例にとり，上記 (2) と (3) の性質が確かに満たされることを確認する．

例 2. $q = 3$ とする[*37]．$h = g^2$ とすると，離散対数問題 $h = g^s$ の解は明らかに $s = 2$ である．また，$f(x,y) = g^{2x} h^y = g^{2x+y}$ である．

- $(k_1, k_2) \in K_0 (= \ker(f)) = \{(k_1, k_2) \mid k_1 + 2k_2 \equiv 0 \pmod 3\}$ は，f により，$g^0 = 1$ に移る．

[*36] 定義 X を持つ関数 f に対して，集合 $f(X)$ を，$f(X) = \{f(x) \mid x \in X\}$ で定義する．

[*37] つまり，$g^3 = 1$ である．

- $(k_1, k_2) \in K_1 = \{(k_1, k_2) \mid k_1 + 2k_2 \equiv 1 \pmod 3\}$ は，f により，g^1 に移る．
- $(k_1, k_2) \in K_2 = \{(k_1, k_2) \mid k_1 + 2k_2 \equiv 2 \pmod 3\}$ は，f により，g^2 に移る．

また，g^0, g^1, g^2 はすべて異なるため，$f(K_1), f(K_2), f(K_3)$ はすべて異なる．

第 5 章

離散対数問題を求解する量子アルゴリズム

4.4 章で説明したとおり，関数 $f(x,y) = g^x h^y$ における核の基底を求めることによって，離散対数問題 $h = g^s$ の解 s を求めることができる．そこで，本章では，離散対数問題を求解する量子アルゴリズム，すなわち，4.2 章で整理した 2 変数全射準同型写像の核の基底を求める際の量子アルゴリズムを解説する．

G を有限可換群とし，G の位数を q とする（q は自然数として，未知の場合も許容し，必ずしも素数とは限らないとする）．

以下では，離散対数問題を以下の 2 つの観点により分類して議論する．

- 群 G の位数 q が既知の場合と未知の場合
- 解 s の大きさに制限がない場合と制限がある場合（解が小さいなど）

表 5.1 にその分類を示す．

表 5.1　4 つの離散対数問題の設定

	位数が既知	位数が未知
解の大きさに制限なし	ケース 1 (5.1 章で解説)	ケース 2 (5.2 章で解説)
解の大きさに制限あり	ケース 4	ケース 3 (5.3 章で解説)

ケース 1 は，暗号に利用される離散対数問題の最も標準的な設定である．ケース 2 と 3 は，離散対数問題の法を合成数 $N = pq$ としたとき，位数である $\mathrm{lcm}(p-1, q-1)$ が未知のケース，つまり，p と q が未知の場合に該当する．このうち，解の大きさに制限がある

離散対数問題については，同問題を経由して素因数分解できることが示されている [14]. これは，離散対数問題 $h \equiv g^x \pmod N$ の解 x を求めることができれば，x を用いて N の素因数 p と q を求めることができるというものであり，その概要については，5.4 章で詳しく説明する.

5.1 位数が既知で解の大きさに制限がない場合

本節では，位数 q が既知で解の大きさに制限がない場合の離散対数問題を求解するアルゴリズムを説明する．表 5.1 でのケース 1 に対応する.

自然数 t を $t = \lceil \log q \rceil$ とする[*38]．つまり，t は $q < 2^t$ を満たす最小の自然数である．4.2 章で整理した 2 変数全射準同型写像の核の生成元を求める量子アルゴリズムは，3 個の量子レジスタを用いて，以下で与えられる．本量子アルゴリズムの回路図を，図 4 で与える.

(量子計算)

ステップ 1： 初期状態を用意する

$$\underbrace{|0\rangle \cdots |0\rangle}_{t \text{ 個}} \underbrace{|0\rangle \cdots |0\rangle}_{t \text{ 個}} |1\rangle$$

ステップ 2： 第 1 量子レジスタ，第 2 量子レジスタに対してアダマール変換を施す.

$$\to \frac{1}{2^t} \sum_{x=0}^{2^t-1} \sum_{y=0}^{2^t-1} |x\rangle |y\rangle |1\rangle$$

ステップ 3： べき乗計算 $h^x g^y$ を適用し，結果を第 3 量子レジスタに書き込む.

$$\to \frac{1}{2^t} \sum_{x=0}^{2^t-1} \sum_{y=0}^{2^t-1} |x\rangle |y\rangle |g^x h^y\rangle$$

ステップ 4： 第 1 量子レジスタ，第 2 量子レジスタに対して逆量子フーリエ変換（回路図上における $\mathrm{QFT}^\dagger$）を作用させる.

ステップ 5： 第 1 量子レジスタ，第 2 量子レジスタを測定する．ここで測定値は，ランダムな $l \in \{0, 1, \ldots, q-1\}$ に対して，l/q の近似値および $ls/q \bmod 1$ の近似値となっている.

[*38] $\lceil x \rceil$ は x より大きい最小の自然数である.

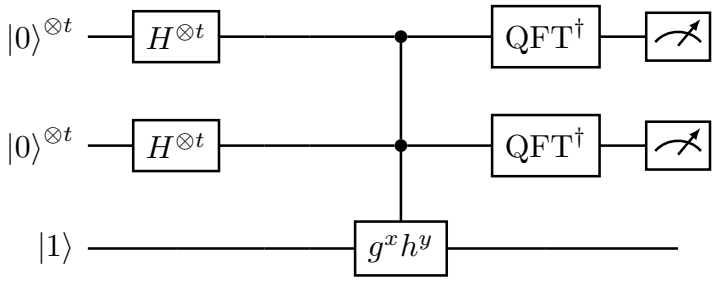

図 4　回路図（離散対数問題を求解する回路）

(古典コンピュータによる処理)
4.2 章で示した方法により，古典コンピュータを用いて，以下の手順により，測定して得られた値から s を復元する．ステップ 5 での測定により，l/q の近似（α とする）と $ls/q \bmod 1$ の近似値（β とする）を得ることができる．測定により得た α, β および既知の q を用いて，以下の計算により，s（と l）を求める．$\alpha' \coloneqq \lfloor \alpha q \rceil$ は高い確率で l となり[*39]，$\beta' \coloneqq \lfloor \beta q \rceil$ は，高い確率で $ls \bmod q$ となる．これより，$\beta'/\alpha' \bmod q = s$ を得る．
具体的なアルゴリズムは以下のとおりである．

ステップ 1：　$\alpha' \coloneqq \lfloor \alpha q \rceil, \beta' \coloneqq \lfloor \beta q \rceil$ を計算する．
ステップ 2：　$s = \beta'/\alpha' \bmod q$ を計算する．

量子計算で測定に用いる量子ビット数は，

- 第 1 レジスタは，t 量子ビット，
- 第 2 レジスタは，t 量子ビット，

をあわせた $2t$ 量子ビットおよび $h^x g^y$ を計算する際に用いる量子ビットの合計値となる．後者は，いずれのケースでも変わらないため，これ以降，測定に用いる量子ビット数のみを考える．

5.2　位数が未知で解の大きさに制限がない場合

本節では，位数 q 自体は未知であるが，その上限値が既知（Q を自然数として，上限値を 2^Q とおく）の場合の量子アルゴリズムを説明する．表 5.1 でのケース 2 に対応する．ベースとなる量子アルゴリズムは，ケース 1（位数が既知）の場合と同一であるが，2 つ

[*39] $\lfloor x \rceil$ は，x を小数点以下四捨五入した整数値である．

変更点がある．1 つ目の変更点は，必要となる量子ビット数である．この量子アルゴリズムで利用する量子ビットは以下で与えられる．

- 第 1 量子レジスタは，$2Q$ 量子ビット
- 第 2 量子レジスタは，Q 量子ビット

つまり，合計で，第 1 量子レジスタ，第 2 量子レジスタ合わせて $3Q$ 量子ビットが必要である．これは，q が既知であったときと比較して，1.5 倍[*40]の量子ビットの増加となる．

2 つ目の変更点は，測定値から解を求める方法である．詳細は割愛するが，ステップ 5 での測定により，$(ls \bmod q)/q$ の近似値（β とする）（同値であるが，ls/q の小数部分の近似値）と l/q の近似値（α とする）を得ることができる．β を連分数展開することにより，$ls \bmod q$ および q を求め，α と q から l を求めるアイデアを用いる．

具体的には，以下の古典計算により，α と β を用いて，s を求める

ステップ 1： α を連分数展開することにより，q および l を求める．
ステップ 2： $\beta' \coloneqq \lfloor \beta q \rceil = ls \mod q$ を計算する．
ステップ 3： $s = \beta'/l \bmod q$ を計算する．

5.3 位数が未知で解の大きさに制限がある場合

本節では位数が未知で解の大きさに制限がある場合の量子アルゴリズムについて説明する．表 5.1 でのケース 3 に対応する．

5.3.1 エケラによる量子アルゴリズム

エケラ [13] により，解が小さい離散対数問題を求解する量子アルゴリズムが提案されている．この論文では，解が小さいということを知っているという前提で議論をしている[*41]．d の上限は既知であり，2^l とする[*42]．さらに，位数 q 自体は未知であるが，その上限は既知であり，2^Q であるとする．

[13] では，

[*40] $t \approx Q$ とした場合
[*41] これ以降，解は d とする．
[*42] 5.2 章では，解に制限のない場合を説明しているが，q は解の自明な上限であるため，ケース 2 はケース 3 の特殊ケースとみなすことができる．

- 第 1 量子レジスタを $2l$ 量子ビット
- 第 2 量子レジスタを l 量子ビット

で構成し，合計で $3l$ 量子ビットで構成される離散対数問題を求解する量子アルゴリズムを示している．なお，解が小さいという制限がない場合は，$3Q$ 量子ビット必要であることに注意されたい（ケース 2)．

この量子アルゴリズムは，$l \ll Q$ のときに，特に有効である．$l \approx Q$ のときにも，有効であるように改良した量子アルゴリズム [14] については，5.3.2 章で説明する．

このアルゴリズムは，測定値から解を求めるのに，連分数展開ではなく，格子理論を利用するという特徴を持つ．格子理論の暗号応用に関しては，[3] の第 4 章に詳しい．

量子計算部分により，測定値として (j, k) が得られたとする．いま，$q > 2^{2l} + (2^l - 1)d$ を満たすとする．基底行列

$$\begin{pmatrix} 4j & 1 \\ 4 \cdot 2^{2l} & 0 \end{pmatrix}$$

として，この行列により張られる格子を $L(j)$ とする．また，$\vec{v} = (4\{-2^l k\}_{2^{2l}}, 0)$ とする．ここで，記号として，$\{u\}_n$ は，u の n に対する余りであるが，$-n/2 \leq \{u\}_n < n/2$ に制限したものとする[*43]．

次に，格子簡約アルゴリズム[*44]を用いて，ベクトル $\vec{v}$ に最も近い $L(j)$ 上の点を求める．これは，$\vec{v}$ との距離が $\sqrt{2} \times 2^l$ よりも小さい $L(j)$ 上の点を求めることに相当する．得られたベクトルの第 2 要素を解 d の候補として，正しい解であるかどうかを確認する．解でない場合には，その次に近い格子 $L(j)$ 上の点を求める．

5.3.2 エケラらによる改良量子アルゴリズム

エケラらは，エケラの量子アルゴリズム [13] を改良することにより，より少ない量子ビットで離散対数問題を求解する量子アルゴリズムを提案している [14]．この論文には，

(a) 離散対数問題の解を分割したうえで，格子理論を利用して，解を求めるという改良
(b) 離散対数問題の解を用いて，素因数分解につなげるという議論

の 2 つが書かれている．(b) は 5.4 章で説明することとし，本節では (a) について詳しく説明する．

[*43] 例えば，$\{4\}_3 = 1, \{5\}_3 = -1$ などである．

[*44] 格子 L の基底が与えられたときに，各基底ベクトルが短くかつ互いに直交に近い L の基底に変換する操作を格子簡約と呼ぶ．その結果，格子に属する短いベクトルを得ることができる．

s を適切に定めたパラメータとする．l' を l/s に近い整数とする． [14] による提案量子アルゴリズムでは，以下の量子ビット数のレジスタを用意する．

- 第 1 量子レジスタを $l + l' \approx l(1 + 1/s)$ 量子ビット
- 第 2 量子レジスタを $l' \approx l/s$ 量子ビット

そのため，合計で，$l(1 + 2/s)$ 量子ビットで十分である．$s = 1$ のときには，5.3.1 章で説明した量子アルゴリズムと同一である．

以下, 具体的に測定値から, 解を求める方法について詳しく説明する．$|\{dj+2^l k\}_{2^{l'+l}}| \leq 2^{l-2}$ を満たすペアを “good pair” (j, k) と呼ぶことにする．ここで，$0 \leq j \leq 2^{l'+l}, 0 \leq k \leq 2^{l'}$ である．量子計算部分を繰り返し用いることにより，good pair を s 個求める．ここで，収集した good pair を，$(j_1, k_1), \ldots, (j_s, k_s)$ とする．

次に，古典アルゴリズムにより，解 d を求める．まず，収集した good pair を用いて，$(s + 1) \times (s + 1)$ 行列を作る．具体的には，格子の基底行列 B を，

$$B = \begin{pmatrix} j_1 & j_2 & \cdots & j_s & 1 \\ 2^{l'+l} & 0 & \cdots & 0 & 0 \\ 0 & 2^{l'+l} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2^{l'+l} & 0 \end{pmatrix}$$

で定義する．B により張られる格子を L とする．具体的には,

$$L = \{B\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{Z}^{s+1}\}$$

で定義される．

ベクトル $\vec{v}$ を，$\vec{v} = (\{-2^l k_1\}_{2^{l'+l}}, \{-2^l k_2\}_{2^{l'+l}}, \ldots, \{-2^l k_s\}_{2^{l'+l}}, 0)$ とする．次に，格子簡約アルゴリズムを用いて，$\vec{v}$ と最も近い格子上のベクトル $\vec{u} \in L$ を求める．実際には，$\vec{v}$ との距離が $\sqrt{s/4 + 1} \times 2^l$ よりも小さい L 上の点を求めることに相当している．$\vec{u}$ の最終要素（第 $s + 1$ 要素）が d の候補となり，正しい解であるかを確認する．解ではない場合は，順次，$\vec{u}$ に近いベクトルの最終要素が解に対応するかを確認する．

5.4 離散対数問題を経由した素因数分解問題の求解

文献 [14] をもとに，離散対数問題の解を使って素因数分解問題を求解するアルゴリズムを示す．これは，古典コンピュータでの処理が可能である．

N を合成数として，N を法とした離散対数問題を考える．ここでは，RSA 型の合成数を考える．つまり，p, q を同じビット長の異なる素数として，$N = pq$ とする．

$g \in \mathbb{Z}_N^*$ を適当に選び，群 G を g から生成される $\mathbb{Z}_N^*$ の部分群とする．いま，

$$X = g^{(N+1)/2} \bmod N$$

とする．ここで，g, N は既知であるので，X は計算可能である．X を変形すると，以下が得られる．

$$\begin{aligned} X &= g^{(N+1)/2} \bmod N = g^{((N-p-q+1)+(p+q))/2} \bmod N \\ &= g^{(p-1)(q-1)/2} g^{(p+q)/2} \bmod N = g^{(p+q)/2} \bmod N \end{aligned} \tag{5.1}$$

式 (5.1) より，この離散対数問題 $X = g^x \bmod N$ の解は $x = (p+q)/2$ となる．なお，$(p+q)/2$ は N や $(p-1)(q-1)$ と比べて非常に小さいこと，p や q と同程度であることに注意されたい．このように，解が小さい離散対数問題を求解できれば，解 $x = (p+q)/2$ から，$p+q$ を得ることができる．ここで，得られた $p+q$ を A とし，$N = pq$ であることから，二次方程式 $x^2 - Ax + N = 0$ を解くことにより，p と q が得られる．以上より，5.3.2 章で示した量子アルゴリズムにより，素因数分解が可能である．

g の位数は，ある自然数 t に対して，$(p-1)(q-1)/t$ と書くことができる．これは，$(p-1)(q-1)/t > (p+q-2)/2$ を満たすことに注意されたい．

p, q をそれぞれ n ビットとしたとき，$2n$ ビットの素因数分解を行うには，通常のショアの量子アルゴリズムであれば，$4n$ 量子ビットの量子レジスタが必要である．（3.3 章を参照．これまでの議論と同様に，これ以外に最終量子レジスタに関する量子ビットが必要である．）

一方，5.3.2 章で示した量子アルゴリズムに必要な量子ビットは，$l(1 + 2/s) = (n + 1)(1 + 2/s)$ である．s を十分大きくとれば，$n + O(1)$ 量子ビットで十分となる．前述のように，通常のショアの量子アルゴリズムでは，$4n$ 量子ビットが必要であったが，エケラらに提案された量子アルゴリズムは，$n + O(1)$ 量子ビットで十分であるため，通常のショアの素因数分解アルゴリズムと比較して，およそ 1/4 の量子ビットで素因数分解が可能である．

5.5 複数離散対数問題を同時に求解する量子アルゴリズム

4.3 章で示した量子アルゴリズムを応用することにより，複数の離散対数問題を同時に求解する問題を考える．G を素数位数 q を持つ有限可換群とし，g を位数 q の元とする．

$h_1, \ldots, h_n \in G$ に対して，

$$h_1 = g^{\alpha_1}, \ldots, h_n = g^{\alpha_n}$$

となる $\alpha_1, \ldots, \alpha_n$ を求める問題を考える．

写像 f を

$$f(x_0, x_1, \ldots, x_n) = g^{x_0} h_1^{x_1} \cdots h_n^{x_n}$$

とする．これを式変形すると，

$$f(x_0, x_1, \ldots, x_n) = g^{x_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n}$$

となる．同様に，

$$\ker(f) = \{(x_0, x_1, \ldots, x_n) \in \mathbb{Z}_q^{n+1} \mid x_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n \equiv 0 \pmod{q}\}$$

となる．前述の量子アルゴリズムにより，$(1, \alpha_1, \ldots, \alpha_n)$ を得ることができる．$(\alpha_1, \ldots, \alpha_n)$ は，ぞれぞれの離散対数問題（$h_1 = g^{\alpha_1}, \ldots, h_n = g^{\alpha_n}$）の解である．

第 6 章

まとめ

本稿では，量子コンピュータに対する従来暗号技術の安全性を議論するために必要となる量子アルゴリズムについて概説した．近年の研究の進展により，量子アルゴリズムによって多項式時間で求解できる問題は可換群の隠れ部分群問題に拡張されていることから，研究の動向把握に繋がるよう，可換群の隠れ部分群問題との関係に着目しながら量子アルゴリズムを解説した．また，正確性を失わない範囲で一般化して解説を行ったことにより，量子コンピュータに対する従来暗号技術の安全性評価の理解が広く促進されることを期待する．

本稿では，全射準同型写像の核を求める問題の量子アルゴリズムを利用して，離散対数問題の求解を経由して素因数分解問題の求解が可能であることを紹介した．この量子アルゴリズムを利用した場合，従来の量子アルゴリズムで想定されていた規模よりも小さい量子コンピュータで素因数分解問題の求解が可能とみられており，この点は，量子コンピュータの実現に伴う現代暗号の安全性評価において非常に重要である．また，本稿では触れなかったが，必要とする量子コンピュータの規模のさらなる量子ビット数の削減が可能であるとの報告もある [17]．こうした素因数分解問題・離散対数問題に関する研究は今後も進展していくとみられることから，量子コンピュータ技術が進展する中で，従来暗号技術の安全性を正確に評価するには，引き続き，素因数分解問題・離散対数問題に対する量子アルゴリズムの効率化に関する研究動向をフォローしていくことが重要である．

参考文献

[1] 木田雅成,『連分数』, 近代科学社, 2022

[2] 國廣 昇, 東京大学工学教程編纂委員会編『基礎系 数学 代数学』, 丸善出版, 2018

[3] ――――・安田雅哉・水木敬明・高安 敦・高島克幸・米山一樹・大原一真・江村恵太, 國廣 昇編『暗号の理論と技術量子時代のセキュリティ理解のために』, 講談社, 2024

[4] 嶋田義皓,『量子コンピューティング 基本アルゴリズムから量子機械学習まで』, オーム社, 2020

[5] 預金取扱金融機関の耐量子計算機暗号への対応に関する検討会,「預金取扱金融機関の耐量子計算機暗号への対応に関する検討会報告書」, 2024 年

[6] CRYPTREC 事務局,「耐量子計算機暗号（PQC）への対応について」, 2025 年

[7] 暗号技術検討会,「暗号技術検討会 2025 年度報告書」, 2025 年

[8] デジタル庁・総務省・経済産業省,「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」, 2022 年

[9] American National Standards Institute X9.62, "Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA)," 1998.

[10] Boudot, Fabrice, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thome, and Paul Zimmermann, "Factorization of rsa-250," 2020 (available at `https://caramba.loria.fr/rsa250.txt`, 2025 年 8 月 5 日).

[11] Diffie, Whitfield, and Martin E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, 22(6), 1976, pp. 644–654.

[12] Duan, Zhao-Chen, Jin-Peng Li, Jian Qin, Ying Yu, Yong-Heng Huo, Sven Hoefling, Chao-Yang Lu, Nai-Le Liu, Kai Chen, and Jian-Wei Pan, "Proof-of-principle Demonstration of Compiled Shor's Algorithm Using a Quantum Dot Single-Photon Source," Optics Express, 28(13), 2020, pp. 18917–18930.

[13] Ekera, Martin, "Modifying Shor' s Algorithm to Compute Short Discrete Logarithms," Cryptology ePrint Archive, Paper 2016/1128, 2016, (available at https://eprint.iacr.org/2016/1128, 2025 年 8 月 5 日).

[14] ————, and Johan Håstad, "Quantum Algorithms for Computing Short Discrete Logarithms and Factoring RSA Integers," in Tanja Lange and Tsuyoshi Takagi, eds. Post-Quantum Cryptography, Springer International Publishing, 2017, pp. 347–363.

[15] Elgamal, Taher, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, 31(4), 1985, pp. 469–472.

[16] Engineering National Academies of Sciences and Medicine, Quantum Computing: Progress and Prospects, The National Academies Press, Washington, DC, 2019.

[17] Gidney, Craig and Ekerå, Martin, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," Quantum, Vol. 5, 2021, p. 433.

[18] Nielsen, Michael A., and Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[19] National Institute of Standards and Technology, "FIPS203, Module-Lattice-Based Key-Encapsulation Mechanism Standard," National Institute of Standards and Technology, 2024 (available at https://csrc.nist.gov/pubs/fips/203/final, 2025 年 8 月 5 日).

[20] ————, "FIPS204, Module-Lattice-Based Digital Signature Standard," National Institute of Standards and Technology, 2024 (available at https://csrc.nist.gov/pubs/fips/204/final, 2025 年 8 月 5 日).

[21] ————,"FIPS205, Stateless Hash-Based Digital Signature Standard," National Institute of Standards and Technology, 2024 (available at https://csrc.nist.gov/pubs/fips/205/final, 2025 年 8 月 5 日).

[22] ————, "Post-Quantum Cryptography," National Institute of Standards and Technology, 2025 (available at https://csrc.nist.gov/projects/post-quantum-cryptography, 2025 年 8 月 5 日).

[23] Rivest, Ronald L., Adi Shamir, and Leonard M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Commun. ACM, 21(2), 1978, pp. 120–126.

[24] Shor, Peter W., "Algorithms for Quantum Computation: Discrete Logarithms

and Factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134.

[25] Simon, Daniel R., "On the Power of Quantum Computation," SIAM Journal on Computing, 26(5), 1997, pp. 1474–1483.

付録 A

数学的準備

本章では，数学的準備を与える．

A.1　群

まず，群について簡単に説明する[*45]．

定義 2 (群)**.** 集合 G と演算 $\cdot$ を考える．以下の条件を満たすとき，$(G, \cdot)$ は群であるという．

- すべての $x, y \in G$ に対して，$x \cdot y \in G$ が成り立つ．
- すべての $x, y, z \in G$ に対して，$x \cdot (y \cdot z) = (x \cdot y) \cdot z$ が成り立つ．
- ある元 $1 \in G$ が存在して，すべての $x \in G$ に対して，$1 \cdot x = x \cdot 1 = x$ が成り立つ．1 を単位元と呼ぶ．
- すべての x に対して，ある y が存在して，$x \cdot y = y \cdot x = 1$ が成り立つ．

以下では，適宜，演算記号 $\cdot$ は省略する．

定義 3 (可換群)**.** 群 $(G, \cdot)$ を考える．すべての $x, y \in G$ に対して，$x \cdot y = y \cdot x$ が成り立つとき，$(G, \cdot)$ は可換群であるという．

定義 4 (有限群)**.** 要素数が有限である群を有限群という．

定義 5 (剰余類群)**.** n を自然数として, 集合 $\mathbb{Z}_n^*$ を, $\mathbb{Z}_n^* \coloneqq \{a \mid 0 \leq a \leq n-1, \gcd(n, a) =$

[*45] 代数（群を含む）に関する初歩的な事柄に関しては，例えば，[2] などを参照されたい

1\} とする．$a, b \in \mathbb{Z}_n^*$ に対して，演算 $\cdot$ を，$a \cdot b = ab \bmod n$ で定義する．このとき，$(\mathbb{Z}_n^*, \cdot)$ を剰余類群と呼ぶ．ここで，$\gcd(a, b)$ は，2 つの自然数 a, b に対する最大公約数を表す．

定義 6 (巡回群)**.** G が単一の元 $g \in G$ を用いて，$G = \{g^i \mid i \in \mathbb{Z}\}$ と記述できるとき，G は巡回群であるという．g により生成される巡回群を $\langle g \rangle$ と書く．

要素数が素数の群は，常に巡回群になる．

A.2 写像

X, Y を集合とする．X の任意の元 x に対して，集合 Y の $f(x)$ がただ一つに定まるとき，f を X から Y への写像という．このとき，$f : X \to Y$ と書く．すべての $x, x' \in X$ に対して，$f(x) = f(x')$ であれば $x = x'$ が成り立つとき，f は単射であるという．すべての $y \in Y$ に対して，$f(x) = y$ となる $x \in X$ が存在するとき，f は全射であるという．写像 f が全射であり，かつ単射であるとき，f は全単射であるという．

次に群の準同型写像を導入する．$(G_1, +), (G_2, \cdot)$ を群とし，f を G_1 から G_2 への写像とする．すべての $x, y \in G_1$ に対して，$f(x + y) = f(x) \cdot f(y)$ が成り立つとき，f は準同型写像であるという．

f が準同型写像であるとき，$\ker(f) = \{x \in G_1 \mid f(x) = 1_{G_2}\}$ を f の核という．ここで，1_{G_2} を群 G_2 の単位元とする．f が準同型写像であるとき，$\mathrm{Im}(f) = \{f(x) \in G_2 \mid x \in G_1\}$ を f の像という．また，$\mathrm{Im}(f)$ を $f(G_1)$ とも書く．ここで，f が全射であるとき，$\mathrm{Im}(f) = G_2$ となる．

定義 7 (全射準同型写像)**.** $(G_1, +), (G_2, \cdot)$ を群とし，f を G_1 から G_2 への準同型写像とする．すべての $x, y \in G_1$ に対して $f(x + y) = f(x) \cdot f(y)$ が成り立つとき，f は準同型写像であるという．特に，f が全射である場合には，f は全射準同型写像であるという．（$f(G_1) = G_2$ であるとき，f は全射である．）

A.3　連分数展開：二進近似からの有理数の復元

3 章以降で述べる古典コンピュータによる処理部分においては，連分数展開が重要である．連分数展開およびその近似を用いた有理数の復元について説明する[*46]．

連分数とは，分母に分数が含まれている分数のことを指し，以下の様な記号を用いる．

$$[a_1, a_2, a_3, a_4] = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

連分数展開による有理数の復元を説明する．r は N 以下の自然数として，l を $0 < l < r$ なる自然数とする．3.2 章で述べる量子アルゴリズムにより，l/r の $2\lfloor \log N \rfloor + 1$ ビット近似[*47]が得られるものとする．

いま，ある実数 ϕ に対して，l/r は，

$$\left| \frac{l}{r} - \phi \right| \leq \frac{1}{2r^2}$$

を満たす有理数であるとする．このとき，l/r は ϕ に対する連分数の収束値である．また，l, r を連分数展開に関するアルゴリズム[*48]を利用して計算可能である．

実際に，l/r の近似値が得られたときに，r と l を求める例を示す．

例 3. r は 4 ビット[*49]とし，l/r の 9 ビット近似が二進数表記により $\phi = 0.010111011$ として[*50]得られたとする．この近似から，l と r を求めることが目的である．

いま，ϕ の値を実際に分数で表現すると，187/512 である．これは，

$$\begin{aligned}\phi &= 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-6} + 0 \times 2^{-7} + 1 \times 2^{-8} + 1 \times 2^{-9} \\ &= \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \frac{1}{2^8} + \frac{1}{2^9}\end{aligned}$$

[*46] 連分数自体に関しては，[1] などを参照されたい．

[*47] l/r の近似値であり，二進数で表記したときに，小数点以下 $2\lfloor \log N \rfloor + 1$ ビットを用いて表現したもの

[*48] 具体的なアルゴリズムは，[1] などを参照されたい．

[*49] つまり，r は 15 以下の自然数である．

[*50] 二進数による小数の表記も十進数の場合と同様に定義される．例えば，十進数表記で 0.503 は，$5 \times 10^{-1} + 0 \times 10^{-2} + 3 \times 10^{-3}$ であるように，二進数表記で 0.101 は，$1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$ で与えられる．

であることによる．ϕ の連分数表記は，連分数展開に関するアルゴリズムを用いて計算すると，

$$\frac{187}{512} = [2, 1, 2, 1, 4, 2, 4]$$

が得られる．系列を前方から途中で打ち切った値は，順に，

$$[2] = \frac{1}{2} \to [2, 1] = \frac{1}{3} \to [2, 1, 2] = \frac{3}{8} \to [2, 1, 2, 1] = \frac{4}{11}$$

で与えられる．いま，r は 4 ビットであるので，$[2, 1, 2, 1]$ により十分精度よく，l/r の近似に成功していることがわかる．これより，$l = 4, r = 11$ が得られる．実際，$4/11$ の二進数表記は，$0.010111010111\cdots$ で与えられ，確かに ϕ は $4/11$ の正しい値と十分に近いことがわかる[*51]．

[*51] $4/11$ は，実際は，小数点以下 010111 を繰り返す巡回小数となる．

付録 B

公開鍵暗号，デジタル署名

素因数分解問題や離散対数問題の困難さに安全性の根拠をおく暗号方式について説明する．問題の定式化および現在知られている古典コンピュータによる最良の解析結果についても説明する．

公開鍵暗号は，鍵生成アルゴリズム (Gen)，暗号化アルゴリズム (Enc)，復号アルゴリズム (Dec) の 3 つから構成される．

鍵生成アルゴリズム Gen は，セキュリティ・パラメータ λ に対して，1^λ を入力とし，公開鍵 pk と秘密鍵 sk を出力する．記号としては，$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ と書く．

暗号化アルゴリズム Enc は，平文 m と公開鍵 pk を入力として受け取り，暗号文 c を出力する．記号としては，$c \leftarrow \mathsf{Enc}(\mathrm{pk}, m)$ と書く．

復号アルゴリズム Dec は，暗号文 c と秘密鍵 sk（と公開鍵 pk）を入力として受け取り，平文 m' を出力する．記号としては，$m' \leftarrow \mathsf{Dec}(\mathrm{sk}, c)$ と書く．

暗号方式としての正しさは，

$$\mathsf{Dec}(\mathrm{sk}, \mathsf{Enc}(\mathrm{pk}, m)) = m$$

で定義される．つまり，「公開鍵で暗号化されたのち，その公開鍵に対応する秘密鍵で復号される場合，もとの平文に戻る」ということを意味する．

デジタル署名は，鍵生成アルゴリズム Gen，署名生成アルゴリズム Sig，検証アルゴリズムは Ver の 3 つから構成される．

鍵生成アルゴリズム Gen は，セキュリティ・パラメータ λ として，1^λ を入力とし，公開の検証鍵 vk と秘密の署名鍵 sk を出力する．記号としては，$(\mathrm{vk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ と書く．

署名生成アルゴリズム Sig は，平文 m と署名鍵 sk（と検証鍵 vk）を入力として受け取り，署名 σ を出力する．記号としては，$\sigma \leftarrow \mathsf{Sig}(\mathrm{sk}, m)$ と書く．

検証アルゴリズム Ver は，平文 m，署名 σ と検証鍵 vk を入力として受け取り，署名が正しいかを検証する．1 の出力は検証に合格したものとし．0 の出力は失敗したものとする．記号としては，$1/0 \leftarrow \mathsf{Ver}(\mathrm{vk}, \sigma, m)$ と書く．

署名方式としての正しさは，

$$\mathsf{Ver}(\mathrm{vk}, \mathsf{Sig}(\mathrm{sk}, m), m) = 1$$

で定義される．つまり，署名鍵で署名生成されたのち，その署名鍵に対応した検証鍵で検証した場合，検証合格である 1 が出力されることを意味する．

B.1 RSA 暗号，ElGamal 暗号，ECDH，ECDSA

ここでは，代表的な暗号方式 RSA 暗号，ElGamal 暗号，鍵共有方式 ECDH，デジタル署名方式 ECDSA を説明する [2].

まず，初等整数論の簡単な事実を示す．

定理 1 (Fermat の小定理)**.** p を素数として，$\gcd(a, p) = 1$ となる自然数 a に対して，$a^{p-1} \bmod p = 1$ が成り立つ．

定理 2. p, q を異なる奇素数とし，$N = pq$ とする．$\gcd(a, (p-1)(q-1)) = 1$ となる自然数 a に対して，$a^{(p-1)(q-1)} \bmod N = 1$ が成り立つ．

定理 2 よりも，少し強い定理が成り立つ．

定理 3. p, q を異なる奇素数とし，$L = \mathrm{lcm}(p-1, q-1)$ とする．$\gcd(a, L) = 1$ となる自然数 a に対して，$a^L \bmod N = 1$ が成り立つ．ここで，$\mathrm{lcm}(a, b)$ は，2 つの自然数 a, b に対する最小公倍数を表す．

RSA 暗号 [23] は，素因数分解の困難さに安全性の根拠をおく暗号である．まず，素因数分解について説明する．

問題 4 (素因数分解)**.** 合成数 N が与えられたときに，N を素因数の積で表現する問題である．特に，RSA 暗号では，N として，同じビット長の異なる 2 つの素数 p, q に対して，$N = pq$ として与えられる．このような合成数を RSA 型の合成数と呼ぶことにする．

素因数分解ができれば，RSA 暗号を解くことが可能である．しかし，この逆，つまり，RSA 暗号が解読できれば素因数分解が可能であるか否かは未解決である．そのため，RSA 暗号の解読と素因数分解という 2 つの問題は，完全な意味での等価性は示されてい

るわけではない.

以下，RSA 暗号のアルゴリズムを紹介する.

RSA 暗号

Gen:

入力： 素数のビット長 n

出力： 公開鍵 (N, e)，秘密鍵 d

ステップ 1： n ビットの 2 つの異なる素数 p, q を生成する．$N = pq$ を計算する.

ステップ 2： $\gcd(e, (p-1)(q-1)) = 1$ となる e を選ぶ.

ステップ 3： $ed \equiv 1 \pmod{(p-1)(q-1)}$ となる d を計算する.

Enc:

入力： 公開鍵 (N, e)，平文 m

出力： 暗号文 c

ステップ 1： $c = m^e \bmod N$ を計算する.

Dec:

入力： 公開鍵 (N, e)，秘密鍵 d，暗号文 c

出力： 平文 m'

ステップ 1: $m' = c^d \bmod N$ を計算する.

多くの仕様では，$e = 65537 (= 2^{16} + 1)$ に設定される．その場合，Gen アルゴリズム中の**ステップ** 1 の p, q の生成において，$p-1, q-1$ が，65537 の倍数ではないことの確認があわせて行われる.

暗号化，復号の処理により，正しく平文が復元されることは，定理 2 に基づいて以下により確認できる．まず，Gen アルゴリズム中の**ステップ** 3 の鍵の生成法より，ある自然数 k が存在して，$ed = 1 + k(p-1)(q-1)$ が成り立つ．$c^d \bmod N$ を計算すると,

$$c^d \bmod N = (m^e)^d \bmod N = m^{ed} \bmod N = m^{1+k(p-1)(q-1)} \bmod N = m$$

となり，確かに，m が復元することが確認できる.

素因数分解問題を多項式時間で求解するアルゴリズムは知られていない．RSA 型の合成数に対する素因数分解問題を求解する最良のアルゴリズムは数体ふるい法であり，その計算量は,

$$\exp\Big(1.923(\log n)^{1/3}(\log\log n)^{2/3}\Big)$$

で与えられる．これまでに素因数分解された RSA 型の合成数は，RSA-250 という，250 桁（829 ビット）の合成数である [10]．現在，RSA 暗号で標準的に利用される合成数は 2048 ビットであり，当分は素因数分解されないと強く信じられている．

次に，離散対数問題の説明を行う．

問題 5 (離散対数問題)**.** p を素数とする．g, h を 1 以上 p 未満の自然数として，$h \equiv g^s \pmod{p}$ となる $s \in \mathbb{Z}$ を求めよ．

問題 6 (一般化された離散対数問題)**.** G を可換群とし，G の位数を n とする．$g, h \in G$ としたときに，$h = g^s$ となる $s \in \mathbb{Z}_n$ を求めよ．

多くの暗号応用では，位数 n として素数に限定する．ただし，5 章の説明では，必ずしも素数には限定しない．

法 p 上で定義される離散対数問題を求解するアルゴリズムの計算量は，p のビット長の準指数関数時間で与えられる．特に，知られている最良のアルゴルズムの計算量は，ある c に対して，

$$\exp\Big(c(\log p)^{1/3}(\log\log p)^{2/3}\Big)$$

で与えられる．安全性を保証するため，2048 ビットの p を利用することが推奨されている．

位数 q の可換群に対する離散対数問題を求解する汎用的なアルゴリズムの計算量は，$O(\sqrt{q})$ で与えられる．安全性を保証するため，256 ビットの q を利用することが推奨されている．実際の仕様では，楕円曲線上で定義された可換群を用いることが多い．

最初に，ElGamal 暗号を説明する．

ElGamal 暗号 [15]

Gen:

ステップ 1： 2048 ビット素数を生成し，p とする．ただし，$p-1$ は 256 ビット素因数 q を持つとする．

ステップ 2： g を位数 q の元とする．(p, q, g) をシステム・パラメータとして，出力する．

ステップ 3： $\mathbb{Z}_q$ の要素 x をランダムに選ぶ．

ステップ 4： $y = g^x \bmod p$ を計算する．y を公開鍵，x を秘密鍵として出力する．

Enc: 平文 $m \in \langle g \rangle \subset \mathbb{Z}_p^*$ とする．

ステップ 1： $\mathbb{Z}_q$ の要素 r をランダムに選ぶ．

ステップ 2： $(C_1, C_2) = (g^r \bmod p, my^r \bmod p)$ を暗号文とする．

Dec:

ステップ 1： $m' = C_2/C_1^x \bmod p$ により復号する．

アルゴリズム Gen 中の**ステップ** 1 において, 実際は, 256 ビット素数 q を選び, $p = kq+1$ が素数となるように自然数 k を選ぶことにより行われる．

アルゴリズム Dec により正しく平文 m が復元できることを確認する．

$$C_1^x \bmod p = (g^r)^x \bmod p = g^{rx} \bmod p = (g^x)^r \bmod p = y^r \bmod p$$

が成り立つ．これより，

$$m' = C_2/C_1^x \bmod p = my^r/y^r \bmod p = m$$

が成り立つため，平文 m が復元される．

次に Diffie–Hellman 鍵共有 [11] を説明する．アリスとボブの二人で鍵共有をしたいとする．共有した鍵は，共通鍵暗号で用いる鍵などに利用する．

Diffie–Hellman 鍵共有

セットアップ：

- 素数 p，$p-1$ の約数である素数 q，位数 q の元 g をシステム・パラメータとする．

アリス：

ステップ 1： $\mathbb{Z}_q$ の要素 a をランダムに選ぶ．

ステップ 2： $A = g^a \bmod p$ を計算し，A をボブに送る．

ボブ：

ステップ 1： $\mathbb{Z}_q$ の要素 b をランダムに選ぶ．

ステップ 2： $B = g^b \bmod p$ を計算し，B をアリスに送る．

その後，以下の計算をおのおの行い共有鍵を得る．

アリス：

$X = B^a \bmod p$ を計算する

ボブ：

$Y = A^b \bmod p$ を計算する

同一の鍵が共有できたことは，以下により確認できる．

$$X = B^a \bmod p = (g^b)^a \bmod p = g^{ab} \bmod p = (g^a)^b \bmod p = A^b \bmod p = Y$$

楕円曲線上で定義された群を用いた Diffie–Hellman 鍵共有方式は ECDH と呼ばれる．簡単に楕円曲線上で定義された群について説明する．有限体 F 上で定義された楕円曲線上の点の集合を考える．この集合に無限遠点を加えた集合を $E(F)$ とする．$E(F)$ 上の 2 点に適切な演算 $+$ を導入することにより，$(E(F), +)$ は有限可換群となる．点 G の a 倍点を $[a]G$ と書くことにする．

この設定のもとで，ECDH を説明する．

ECDH

セットアップ：

- $E(F)$ の位数は素数 q とする．
- 点 R を位数 q の点とする．

アリス：

ステップ 1： $\mathbb{Z}_q$ の要素 a をランダムに選ぶ．

ステップ 2： $A = [a]R$ を計算し，点 A をボブに送る．

ボブ：

ステップ 1： $\mathbb{Z}_q$ の要素 b をランダムに選ぶ．

ステップ 2： $B = [b]R$ を計算し，点 B をアリスに送る．

その後，以下の計算をおのおの行い共有鍵を得る．

アリス：

点 $X = [a]B$ を計算する．

ボブ：

点 $Y = [b]A$ を計算する．

同一の鍵が共有できることは，以下により確認できる．

$$X = [b]A = [b]([a]R) = [ab]R = [a]([b]R) = [a]B = Y$$

次に，楕円曲線に基づくデジタル署名方式 ECDSA [9] の説明をする．

ECDSA

セットアップ：

- 有限体 $\mathbb{F}_p$ 上で定義された楕円曲線 E 上の位数 n の点 R をとり，公開パラメータを (E, p, R, n) と設定する．

Gen:

ステップ 1： 署名鍵 d を $\mathbb{Z}_n$ からランダムに選ぶ．

ステップ 2： 検証鍵を $Q = [d]R$ とする．

Sig:

入力： メッセージ $m \in \{0,1\}^*$

出力： 署名 (r, σ)

ステップ 1： $h = H(m)$ とする．ここで，H はハッシュ関数とする．

ステップ 2： $\mathbb{Z}_n$ の要素 k をランダムに選ぶ．

ステップ 3： 楕円曲線上で点 $[k]R$ を計算する．点 $[k]R$ の x 座標を x_1 とする．

ステップ 4： $r = x_1 \bmod n$ を計算する．$r = 0$ であれば，k を選択しなおす．

ステップ 5： $\sigma = (h + rd)/k \bmod p$ を計算する．$\sigma = 0$ であれば，k を選択しなおす．

Ver:

ステップ 1： $h = H(m)$ を計算する．

ステップ 2： $w = \sigma^{-1} \bmod n$ を計算する．

ステップ 3： $u_1 = hw \bmod n, u_2 = rw \bmod n$ を計算する．

ステップ 4： $P = [u_1]R + [u_2]Q$ を計算し，P の x 座標を x_1 とする．

ステップ 5： $r = x_1 \bmod n$ であれば，1 を出力する．

正しく署名が生成された場合は，常に検証に合格し，1 が出力されることを確認する．アルゴリズム Ver の**ステップ** 4 において，

$$\begin{aligned} P &= [u_1]R + [u_2]Q = [u_1]R + [u_2][d]G = [u_1 + u_2 d]R \\ &= [zw + rwd]R = [w(z + rd)]R = [(z + rd)/\sigma]R = [k]R \end{aligned}$$

となる．これより，正しく署名を生成した場合には，常に検証で 1 が出力される．

離散対数問題を求解できれば，ElGamal 暗号，ECDH，ECDSA なども解くことができることに注意されたい．

付録 C

隠れ部分群問題

本章では，隠れ部分群の定義を与えるとともに，この問題を求解する汎用的な量子アルゴリズムを説明する．

C.1 隠れ部分群問題

まず，3 章で述べた隠れ部分群問題の定義を再掲する．

定義 8 (隠れ部分群問題)**.** 有限生成群 H と有限集合 X に対して，H から X への関数 f がオラクルとして与えられており，すべての $x, y \in H$ に対して，

$$f(x) = f(y) \iff x - y \in K$$

が成り立つとする．ここで，K は H の未知の部分群である．このとき，K を定めよ．

X は実質的に，$f(H) = \{f(x) \mid x \in H\}(\subseteq X)$ と制限しても問題ないため，これ以降は，関数 f，有限生成群 H，H の部分群 K を中心に考える．

以下，有限生成群に関して知られている事実を記載する．

定理 4 (有限生成可換群の基本定理)**.** 群 G が有限生成可換群であるならば, $e_1 > 1, e_i | e_{i+1}$ となるような自然数 $e_1, e_2, \ldots, e_s$ と非負整数 r が存在し，

$$G \cong \mathbb{Z}/e_1\mathbb{Z} \times \cdots \times \mathbb{Z}/e_s\mathbb{Z} \times \mathbb{Z}^r$$

となる．

隠れ部分群問題の困難さに関して，以下の結果が知られている．

- G が有限生成可換群であれば，量子コンピュータにより多項式時間で求解可能（Simon の問題 [25]，素因数分解問題，離散対数問題など）．
- G が非可換群（例えば，対称群や二面体群）のとき，量子コンピュータを用いても多項式時間で求解可能な量子アルゴリズムは知られていない．
  - グラフ同型性判定問題は，G が対称群の場合に相当している．
  - G が二面体群であれば，準指数関数時間で求解可能である．ある種の格子問題に関連がある．

C.2 隠れ部分群問題に対する標準的な量子アルゴリズム

ユニタリ変換 U_f は，関数 f に対して，第 1 量子レジスタの値を引数として受け取り，計算した値を第 2 量子レジスタに書き込むものとする．いま，$U_f |h\rangle |0\rangle = |h\rangle |f(h)\rangle$ を実行する量子オラクルが与えられているとする．

以下の量子アルゴリズムは，隠れ部分群問題に対する標準アルゴリズム (Standard Algorithm) と呼ばれる．

ステップ 1： 初期状態 $|0\rangle |0\rangle$ を準備する．

ステップ 2： フラットな量子重ね合わせを作成

$$|0\rangle |0\rangle \longrightarrow \frac{1}{\sqrt{|H|}} \sum_{x \in H} |x\rangle |0\rangle$$

ステップ 3： ユニタリ変換 U_f を適用

$$\frac{1}{\sqrt{|H|}} \sum_{x \in H} |x\rangle |0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{|H|}} \sum_{x \in H} |x\rangle |f(x)\rangle$$

ステップ 4： 第 1 量子レジスタを逆量子フーリエ変換する．

ステップ 5： 第 1 量子レジスタを測定する．

量子計算終了後，得られた測定値から，古典計算により，解（K）を導き出すことが必要であることに注意が必要である．

付録 D

量子アルゴリズム

D.1　量子回路における基本ゲート

古典コンピュータでは，古典ビットに対して，AND や OR などの古典ゲートを操作することにより計算が行われており，論理回路を用いて計算を記述可能である．同様に，量子コンピュータでは，量子ビットに対して，量子ゲートを操作することで計算を行うものであり，その計算は量子回路により記述可能である．

本章では，まず，量子計算を理解するうえで重要となる量子ゲートと量子回路について説明する．量子回路に関する一般的な議論に関しては，[3, 4, 16, 18] などを参照されたい．また，本章の記述の多くは，[3] をもとにしている．

以下，i は虚数単位であり，$\sqrt{-1}$ である．$\mathbb{C}$ は複素数の集合とする．

D.1.1　量子ビット

量子ビットは，同時に 0 と 1 の両方の状態を持つ（重ね合わせ）ことができ，$|0\rangle, |1\rangle$ という記号を用いて表される．$\alpha, \beta \in \mathbb{C}$ とするとき，量子ビットの状態 Ψ は，

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

で表記される．α と β はどの程度の重みで 0 状態と 1 状態が重ね合わさっているかを表しており，この状態を測定すると，それぞれの量子ビットが状態 $|0\rangle$ と $|1\rangle$ になる確率は $|\alpha|^2$ と $|\beta|^2$ となる．

こうした量子ビットはベクトルを用いて表すことができる．

定義 9 (量子ビットのベクトル表現)**.** $|0\rangle, |1\rangle$ を以下のベクトルにより表現する.

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

ベクトル表現を用いると，量子状態の重ね合わせは，次のように記述可能である.

$$\alpha|0\rangle + \beta|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$$

ただし，$\alpha, \beta \in \mathbb{C}$ であり，$|\alpha|^2 + |\beta|^2 = 1$ という制約を満たす.

1 量子ビットを 2 つ並べた 2 量子ビットの場合は，$00, 01, 10, 11$ の 4 通りの状態の重ね合わせを取りうる．それぞれに対応する量子状態を $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ と記述する．第 1 量子ビットと第 2 量子ビットの役割を区別したい場合には, $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$ とも記述する[*52].

D.1.2 量子ゲート

量子ビットの状態がベクトルで表されるため，量子ゲートの操作はベクトルに作用する行列で表される．そのため，量子回路は線形性を持つ．つまり，U を量子回路による演算とすると，すべての α, β に対して,

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle$$

が成り立つ.

量子ゲートには作用させる前後で全状態の確率の和が保存されなくてはならないという制約がある（1 量子ビットの場合では，$|\alpha|^2 + |\beta|^2 = 1$）．これは作用させたベクトルの長さを変えないことと同値であり，このような性質を満たす複素行列をユニタリ行列という．量子ゲートはすべて，このユニタリ行列で構成される必要がある．量子回路を理解するうえで重要となるユニタリ行列の定義を示す.

定義 10 (ユニタリ行列)**.** U を複素正方行列[*53]とし，$U^\dagger$ を U の随伴行列[*54]とする．$UU^\dagger = U^\dagger U = I$ が成り立つとき，U はユニタリ行列であるという．ここで，I は単位行列とする.

[*52] 3 量子ビット以上の場合も同様に記述できる.

[*53] 複素数を成分として持つ正方行列.

[*54] 各成分の複素共役を取ったうえで，行列全体を転置した行列．ここで，a, b を実数として，複素数 $a + b\mathrm{i}$ の複素共役は，$a - b\mathrm{i}$ で与えられる.

ユニタリ行列は，以下の特徴を持つ．

- U の逆行列は $U^\dagger$ である．
- U のすべての固有値の絶対値は 1 である．
- U は等長変換である．つまり，すべての $\boldsymbol{x}$ に対して，$\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$ が成り立つ．
- 二つのユニタリ行列 U_1, U_2 の積 U_2U_1 もユニタリ行列である．
- U_1, U_2 をユニタリ行列とすると，U_2U_1 の逆行列は，$U_1^\dagger U_2^\dagger$ である．

量子計算では，量子状態を操作する量子ゲートを組み合わせて量子回路を作成する．つまり，初期状態 $|0\rangle$ の量子状態を準備し，量子ゲートと測定という操作を組み合わせることで計算が実行される．以下では，本稿で説明する量子アルゴリズムの理解に必要となる代表的な量子ゲートの説明を行う．

まず，1 量子ビットゲート[*55]として，アダマールゲート H，NOT ゲート X（を含むパウリゲート），位相シフトゲート $P(\theta)$ の説明を行う．アダマールゲートは，量子ビットの重ね合わせ状態を作成するときによく使用されるゲートであり，NOT ゲートは，量子ビットの状態を反転させるゲート，位相シフトゲートは，量子状態の回転を行うゲートである．さらに，位相シフトゲートの特殊例として，S ゲート，T ゲートの説明も合わせて行う．さらに，2 量子ビットゲートの代表例である制御 NOT ゲート（英語では，Controled-NOT ゲートと表記し，以下，CNOT ゲートと記載する）と制御シフトゲート，および 3 量子ゲートの代表例であるトフォリゲートを説明する．

以下で紹介するすべてのゲートを，3 章以降の説明で用いているわけではない．しかし，実際の量子コンピュータによる回路構成まで考えた場合，トフォリゲートなどの複数量子ビットを操作できる便利なゲートを用いて構成し，最終的に実際の量子コンピュータが扱いやすいパウリゲートにまで分割した上で，実行されることが多い．そのため，本章では，量子回路の動作を理解する上で必須のゲートについて紹介する．

[*55] 1 量子ビットゲートは，1 量子ビットに作用するゲートの総称である．

アダマールゲート

アダマールゲート H の演算は,

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

で定義される．状態 $|0\rangle$ と状態 $|1\rangle$ に対して，アダマールゲート H を作用させると，状態がそれぞれ $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ と $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ に遷移することを意味し，行列表現では，$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ となる．これは,

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\end{pmatrix} - \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

であることにより，確認できる．

図 5 に回路図の例を与える．

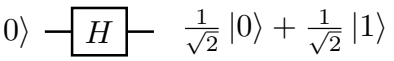

図 5　回路図の例：アダマールゲート

この例では，量子状態 $|0\rangle$ はアダマールゲート H によって，状態 $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ に遷移することを示している．

多くの量子アルゴリズムでは，最初の計算ステップで量子重ね合わせ状態の生成が行われ，その生成には一般的に，アダマールゲートが利用される．実際に，重ね合わせ状態が生成できることを確認する．

簡単のため，2 量子ビットの場合を考える．1 量子ビット目，2 量子ビット目は，それぞれ，$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ と状態遷移する．2 量子ビットが遷移した状態 $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ は，整理すると，$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ と記述することができる．さらに，2 進数表記であるとみなし，$|00\rangle, |01\rangle, |10\rangle, |11\rangle$ をそれぞれ，$|0\rangle, |1\rangle, |2\rangle, |3\rangle$ と

表記できる．以上をまとめると，

$$|0\rangle\,|0\rangle \xrightarrow{H\otimes H} \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) = \frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle) = \frac{1}{2}(|0\rangle+|1\rangle+|2\rangle+|3\rangle)$$

となり，アダマールゲート H によって，2 量子ビットの 4 つの状態 $|00\rangle, |01\rangle, |10\rangle, |11\rangle$（もしくは，$|0\rangle, |1\rangle, |2\rangle, |3\rangle$）を生成することがでわかる．$n$ 量子ビットに対しても，同様の議論が可能である．実際，アダマールゲートを利用することにより，2^n 個の量子状態の重ね合わせの生成が可能であり，

$$\underbrace{|0\rangle\cdots|0\rangle}_{n\text{ 個}} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$$

となる．

NOT ゲート

NOT ゲート X は，

$$X\,|0\rangle = |1\rangle$$
$$X\,|1\rangle = |0\rangle$$

で定義される．ビット b のビット反転を $\bar{b}$ とすると[*56]，$b \in \{0,1\}$ に対して，$X\,|b\rangle = \left|\bar{b}\right\rangle$ と書くことが可能である．また，NOT ゲートを含む以下の（X, Y, Z）で構成されるゲートはパウリゲートと呼ばれ，量子状態をある種の反転させる作用を持つ．ここで，I は単位行列であり，演算としては恒等演算を意味する．

$$I \coloneqq \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},\quad X \coloneqq \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix},\quad Y \coloneqq \begin{pmatrix}0 & -\mathrm{i}\\ \mathrm{i} & 0\end{pmatrix},\quad Z \coloneqq \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

行列 I, X, Y, Z の間には，以下の関係がある．

$$X^2 = Y^2 = Z^2 = I,$$
$$XY = -YX = \mathrm{i}Z, YZ = -ZY = \mathrm{i}X, ZX = -XZ = \mathrm{i}Y$$

位相シフトゲート

位相シフトゲート $P(\theta)$ は次で定義される．

$$P(\theta)\,|0\rangle = |0\rangle$$
$$P(\theta)\,|1\rangle = \exp(\mathrm{i}\theta)\,|1\rangle$$

[*56] $\bar{0} = 1, \bar{1} = 0$ である．

行列表現は,

$$P(\theta) \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \exp(\mathrm{i}\theta) \end{pmatrix}$$

で与えられる．$(P(\theta))^k = P(k\theta)$ が成り立つ．

次に，位相シフトゲートの特殊例として，回路構成において重要な S ゲートと T ゲートを説明する．

S ゲートは,

$$S \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \exp(\pi\mathrm{i}/2) \end{pmatrix} = P\left(\frac{\pi}{2}\right)$$

で定義される．ここで，明らかに $S^2 = Z, S^4 = I$ である．T ゲートは,

$$T \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \exp(\pi\mathrm{i}/4) \end{pmatrix} = P\left(\frac{\pi}{4}\right)$$

で定義される．ここで，$T^2 = S,\ T^4 = Z,\ T^8 = I$ が成り立つ．

CNOT ゲート

CNOT ゲートは，2 入力 2 出力の量子ゲートであり[*57]，入力のうちの 1 つは制御ビット，もう 1 つは標的ビットと呼ばれる．制御ビットの値に応じて，標的ビットに NOT 演算の実行を制御することができる．具体的には，制御ビット c が $c = 0$ のときは，標的ビットには何も作用させず，$c = 1$ のときは，標的ビットの値を反転させる．ケットによる表記を用いると，CNOT ゲートは，以下で表される．

$$|c\rangle\,|t\rangle \mapsto |c\rangle\,|t \oplus c\rangle$$

実際, $c = 0$ のときには, $|0\rangle\,|t\rangle \mapsto |0\rangle\,|t \oplus 0\rangle = |0\rangle\,|t\rangle$ となり, $|1\rangle\,|t\rangle \mapsto |1\rangle\,|t \oplus 1\rangle = |1\rangle\,|\bar{t}\rangle$ となる．

制御位相シフトゲート

次に，制御位相シフトゲートを紹介する．制御位相シフトとは，制御ビットが 0 のときは何も作用させずに，1 のときは制御位相シフト $P(\theta)$ を行うものである．ケットによる表記は，以下で与えられる．

$$|00\rangle \mapsto |00\rangle\,,\quad |01\rangle \mapsto |01\rangle\,,\quad |10\rangle \mapsto |10\rangle\,,\quad |11\rangle \mapsto \exp(\mathrm{i}\theta)\,|11\rangle$$

制御位相シフトゲートにおいて，第 1 ビットと第 2 ビットは対等な関係にある．どちらを制御ビットとするか標的ビットとするかを意識する必要はない．

[*57] 量子回路においては，常に入力数と出力数は等しい．

トフォリゲート

トフォリゲートは，3 入力 3 出力の量子ゲートであり，入力 3 ビットのうち 2 ビットが制御ビット，残りの 1 ビットが標的ビットである．制御ビットを c_1, c_2 とし，標的ビットを t としたとき，トフォリゲートでは，c_1 と c_2 がともに 1 である場合のみに，t をビット反転する．ケットによる表記は，

$$|c_1\rangle |c_2\rangle |t\rangle \mapsto |c_1\rangle |c_2\rangle |t \oplus (c_1 \wedge c_2)\rangle$$

で与えられる．

D.2 量子フーリエ変換

素因数分解問題や離散対数問題などの量子コンピュータによって多項式時間で求解可能な問題は，その量子アルゴリズム中で，いずれも量子フーリエ変換（QFT：Quantum Fourier Transform）を利用している．量子フーリエ変換は，量子ビットに一連の量子ゲートを適用することで，入力状態をフーリエ変換するものである[*58]．これにより，複雑な計算を並列的に処理可能となり，計算の高速化をはかることができる．3 章以降で示す量子アルゴリズムでは，量子フーリエ変換が本質的な役割を果たす．

定義 11. 量子フーリエ変換 は，正規直交基底 $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ に対して，

$$|j\rangle \xrightarrow{\text{QFT}} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(2\pi \mathrm{i} jk/N) |k\rangle \tag{D.1}$$

で定義される．

次に，重ね合わせ状態に対する作用を説明する．N 次元複素数ベクトル $(x_0, \ldots, x_{N-1})$ を入力に対して，

$$y_k \coloneqq \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp(2\pi \mathrm{i} jk/N)$$

とする．ただし，$0 \leq k \leq N-1$ である．このとき，重ね合わせ状態 $\sum_{j=0}^{N-1} x_j |j\rangle$ に対

[*58] フーリエ変換とは，時間領域の信号を周波数成分に分解する手法である．これにより，信号に含まれる周波数成分とその強度を知ることができる．

する量子フーリエ変換は，

$$\sum_{j=0}^{N-1} x_j \left|j\right\rangle \xrightarrow{\mathrm{QFT}} \sum_{k=0}^{N-1} y_k \left|k\right\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1}\sum_{j=0}^{N-1} x_j \exp(2\pi \mathrm{i} jk/N) \left|k\right\rangle$$

で表される．この変換は，ユニタリ変換である．

量子フーリエ変換の逆変換を，逆量子フーリエ変換（$\mathrm{QFT}^\dagger$）と呼ぶことにする．逆量子フーリエ変換は，定義 11 と同じ記号のもとで，

$$\left|j\right\rangle \xrightarrow{\mathrm{QFT}^\dagger} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(-2\pi \mathrm{i} jk/N) \left|k\right\rangle \tag{D.2}$$

で定義される．逆量子フーリエ変換は，量子フーリエ変換と同じコストで実現される．

D.3 量子フーリエ変換を行う効率的な回路

D.2 章で定義した量子フーリエ変換は，任意の N に対して定義されるが，実装を考慮した場合には，n を自然数として，$N = 2^n$ に限定することが多い．そのため，以降は，$N = 2^n$ とし，ケット $\left|0\right\rangle, \left|1\right\rangle, \ldots, \left|2^n - 1\right\rangle$ を基底とする．j の 2 進数表現 $j = j_1 j_2 \cdots j_n$ を用いて，式 (D.1) を変形すると，以下の別表現が得られる．なお，$0.j_1 j_2 \cdots j_n$ は，$j_1/2 + j_2/2^2 + \cdots + j_n/2^n$ を表す．

$$\left|j_1 \cdots j_n\right\rangle \rightarrow \frac{(\left|0\right\rangle + \exp(2\pi \mathrm{i} 0.j_n)\left|1\right\rangle)(\left|0\right\rangle + \exp(2\pi \mathrm{i} 0.j_{n-1}j_n)\left|1\right\rangle)\cdots(\left|0\right\rangle + \exp(2\pi \mathrm{i} 0.j_1 \cdots j_n)\left|1\right\rangle)}{2^{n/2}}$$

$n(n+1)/2$ 個のゲートにより実現される量子フーリエ変換の実装方法を紹介する．ここでは，$n = 4$ の場合を例に説明する．図 6 に，$n = 4$ の場合の量子フーリエ変換を行う回路を示す．

例 4 (回路構成（$n = 4$ の場合の例）)．R_k ゲートを

$$R_k \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & \exp\bigl(2\pi \mathrm{i}/2^k\bigr) \end{pmatrix}$$

とする[*59]．回路図における横線はそれぞれ 1 つの量子ビットに対応している．回路の左から順に実行される．まず，第 1 量子ビットに対して，アダマールゲート H が作用され

[*59] 位相シフトゲートを用いると，$R_k = P(2\pi/2^k)$ で表される．

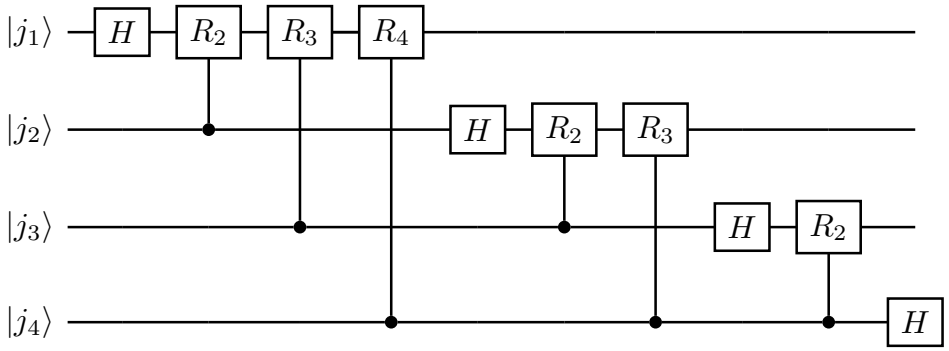

図 6　回路構成（$n = 4$ の場合の例）

る．つぎに，第 2 量子ビットを制御ビット，第 1 量子ビットを標的ビットとして，制御 R_2 ゲートが作用される．（ここで，図中の • は制御ビットとして作用することを表す.）ついで，第 3 量子ビットを制御ビット，第 1 量子ビットを標的ビットとして，制御 R_3 ゲートが作用される．このように，順に，左から右に順番にゲートが実行され，最後に，第 4 量子ビットに対して，アダマールゲートが作用されることにより，量子フーリエ変換全体が行われる．

最終状態は，

$$\frac{(|0\rangle + \exp(2\pi \mathrm{i} 0.j_1 j_2 j_3 j_4)\,|1\rangle) \cdots (|0\rangle + \exp(2\pi \mathrm{i} 0.j_3 j_4)\,|1\rangle)(|0\rangle + \exp(2\pi \mathrm{i} 0.j_4)\,|1\rangle)}{2^{n/2}}$$

であるため，状態の交換を行い順番を戻す必要がある．具体的には，第 4 量子ビットと第 1 量子ビットの状態を入れ替え，第 3 量子ビットと第 2 量子ビットの状態を入れ替える必要がある．