4.1 超ロバスト量子計算

今井浩

情報理工学系研究科コンピュータ科学専攻

要旨

本サブプロジェクトでは，量子状態のデコヒーレンスと操作エラーに基づく計算困難性を克服する研究と，デコヒーレンスによりもたらされる状態を活用する研究との両面から，超ロバスト量子計算について研究する．さらに，量子計算と位相的構造との関係を調べ，新たなロバスト性を確立することを目指している．本報告では，主に今年度遂行したシミュレーションによる量子計算のエラーに対する耐性の研究成果について述べる．

1　はじめに

量子コンピュータは，量子力学原理に基づいて動作するコンピュータである．すなわち，量子状態を内部での情報表現として用い，ある量子状態を他の量子状態に変換する量子的操作を計算手段とし，そして量子測定を情報獲得法としたものである．理論的には素因数分解を既存コンピュータより超高速に行えることが示され，現代の RSA 暗号など公開鍵暗号系のセキュリティに強いインパクトを与えているものの，実現はまだ先だと思われている．その一因は，量子状態が脆く，外界と作用して生じるデコヒーレンスエラーや，計算での操作エラーが存在する中で，ロバストで正しい計算ができる方式・解析が行われていないことにある．

本研究では，超ロバスト計算可能な量子計算のため，まずシミュレーションによる解析を行い理論的に発展させていくとともに，一方で位相的アプローチによる新しいロバスト性の確立も目指す．本報告では，前者を主に今年度の研究成果について述べる．以降，この節では，後者に関する研究計画を記しておく．

これまでに，結び目理論・組み紐理論と量子計算・量子情報との関連が指摘されており，量子計算過程の基本ゲートに対応するユニタリ変換するところを組み紐群と対応させることや，量子状態のもつれ (entanglement) の度合いを組み紐群の不変量と対応させること，結び目の不変量である Jones 多項式を量子計算で計算することが調べられている．

本サブプロジェクトでは，これをさらにより広くトポロジーという概念の持つロバスト性によって，量子計算における新たなロバスト性を確立することを目指している．これは量子計算における量子エラーであるデコヒーレンスや操作エラーを直接的にモデル化してエラーへの耐性を確保する研究方法 (上の分類でいうところの前者; 本報告の次節以降の内容) とは別の視点から，量子計算の新ロバスト性を研究するものである．そのためには，こうした計算に関するトポロジーに絡んだ研究に大きな役割を果たしている，位相に関する組合せ的手法の研究が必要である．

そこで，本サブプロジェクトでは，可能であれば COE によって学生を雇用し，その者に組合せ的手法を軸に位相的手法の研究を遂行させ，基礎理論と同時に計算システムを開発させることを計画している．これにより，量子計算での新ロバスト性確立を目指すサブプロジェクトの土台となる研究成果を産み出すとともに，そこでのソフトウェアとそれをプレゼンテーションすることによってこのプロジェクトの特徴である位相的アプローチをわかりやすく示すことも可能となる．

2 Shor の素因数分解量子アルゴリズム — 全量子計算シミュレーション

1994 年, P. W. Shor は素因数分解を多項式時間でおこなう量子アルゴリズムを発表した. しかし, 実際には L ビットの数を因数分解する量子回路を構築するには $5L+6$ 量子ビットと多数必要であった. したがって, 現存する量子計算機や, 古典計算機によるシミュレーションでは因数分解できる数の範囲は限られていた. 特に, この量子ビット数を使い, かつアルゴリズムの中で最も時間がかかる部分であるモジュラー計算での巾乗の部分については, これまで十分なサイズで量子アルゴリズムをそのままシミュレートした結果は存在しなかった.

しかし, 最近になって, S. Beauregard が量子フーリエ変換 (QFT) を応用した量子加算回路を使用することで, また, 量子フーリエ逆変換を半古典的に適用することで, 量子ビットの数を $2L+3$ 個にまで削減した因数分解アルゴリズムに対する改良版量子回路を提案した. 本研究では, まずこれらを組合わせ, これにより, 既存の並列コンピュータの環境でもより大きな数を因数分解できることを可能にした. そして, シミュレーションにより定量的にアルゴリズムを実際に解析することが行った.

本研究では, 上述の回路のデコヒーレンスエラーならびに操作エラーに対する耐性を評価している. まず上述の回路を丹羽らが開発した量子計算シミュレーションシステム上に実装し, 次にそれを分散型並列計算機 SCore III でシミュレートすることにより, 回路の振る舞いを調査する. 加えて, この回路への近似量子フーリエ変換 (AQFT) の適用可能性についても議論する. 得られる結果は近い将来の実機での検証に資するはずである.

その結果, デコヒーレンスエラー率が 10^{-5} 未満かつ操作エラーの標準偏差が 10^{-2} 未満ならば, 上述の回路は有効に動作することがわかった. また, この回路に対し AQFT が有用であることを確認した. さらには, 4 ビット以上の数の因数分解をシミュレートする場合に並列化の効果がみられることも確認した. 並列実行をおこなうノード数が倍になるとシミュレーションの所要時間がほぼ半分になることを実験により確認した.

以下, この成果について計算結果を中心に説明する.

2.1 Shor のアルゴリズムで量子ビットを減らすこと

Shor のアルゴリズムで量子計算特有の利点を活用しているのは, 量子フーリエ変換の部分である. これを用いて, 通常の古典計算では指数時間かかってしまう素因数分解のために必要な位数計算が, 多項式時間で行える. 通常の量子フーリエ変換の回路を図 1 に示す. これは 4 量子ビットの場合で, 実際には $2^4 = 16$ 次元複素ベクトル空間でのフーリエ変換を, 量子重ね合せ状態の上で量子ビット数に関して多項式時間 (この回路の場合には 2 乗の時間) で行う.

しかし, この量子フーリエ変換をそのまま用いて, 素因数分解を解くための位数計算に適用すると, 量子ビット数がかなり多くなる. Shor のアルゴリズムでは, Walsh-Hadamard 変換までも含めた広い意味での量子フーリエを複数回用いるが, 最も後半で用いる位数計算の部分では半古典的に量子測定を途中でいれることによって, 量子並列性は若干犠牲になって逐次的にはなるが, かなり量子ビットを削減することができる.

さらに, 前段で用いるフーリエ変換において, 近似的にフーリエ変換を行うことが考えられる. すなわち, 図 1 の回路図で離れた量子ビットの間の操作をする際は, もともと相対的に小さな操作しかせず, またそれを実際に量子コンピュータで行うことは相対的に難しいことになるので, それを図 2 のように省略して近似するというものである. 注意すべき点は, これによってフーリエ変換の結果そのものは近似的なものとなるが, 最終的な位数計算のためにはその上で測定を行うので, この近似が影響するのは測定確率であって, 解そのものではないということである. 実際に, 素因数分解では, より小さな数への分解が与えられればそれが正し

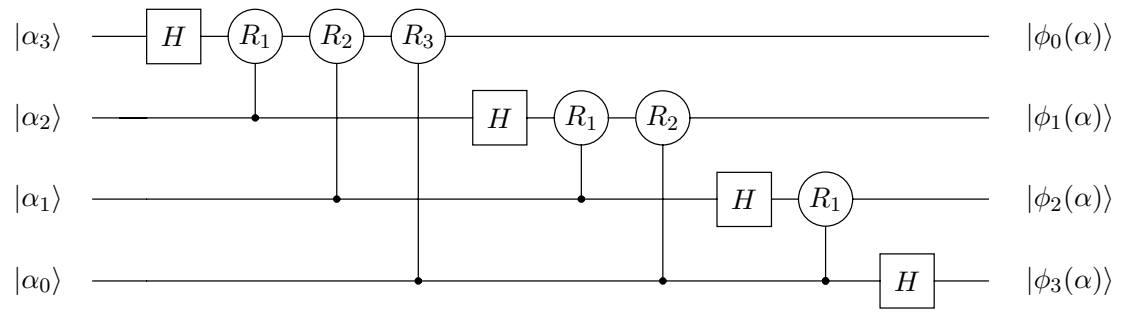


図 1: 4 量子ビットに対する量子フーリエ変換回路

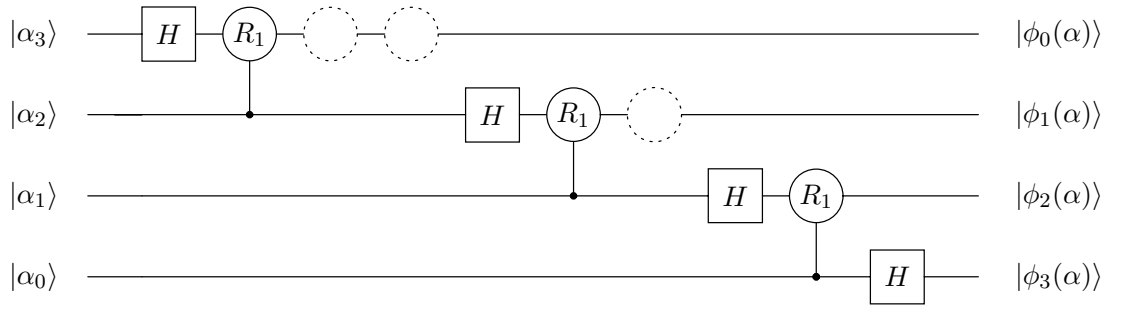


図 2: 近似次数 $m = 2$ の近似量子フーリエ変換回路. 点線で書かれている円が削除されたゲートを表わす.

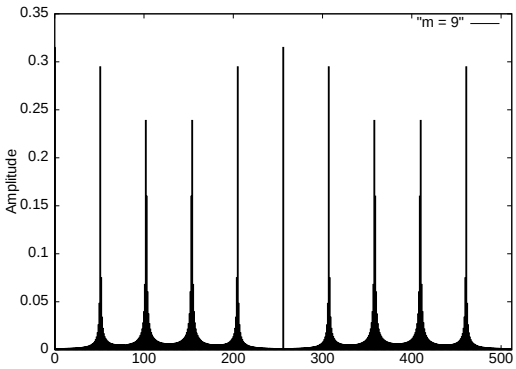
いことは容易に確認できるので，この近似はあくまで成功確率が低くなりうるという点についてである．

このような技術をすべて組合わせることにより，元の Shor のアルゴリズムをそのままで解析すると L ビットの数を因数分解するのに量子ビットとして $5L+6$ 必要であったのを，$2L+3$ まで落すことができる．量子ビットの数を減じることができても，回路の深さがそれ以上に深くなると，全体としてはエラー耐性の点でも問題が生じる可能性があるが，本技法では深さについてさほどの増大を生じておらず，実際に量子コンピュータで実現する際も有望なものである．また，現コンピュータで量子計算をシミュレートする場合，量子ビット数に対して指数的な次元での線形計算が必要であるが，この量子ビットを減らすことにより全ての量子計算の部分を現コンピュータでシミュレートすることが可能になる．

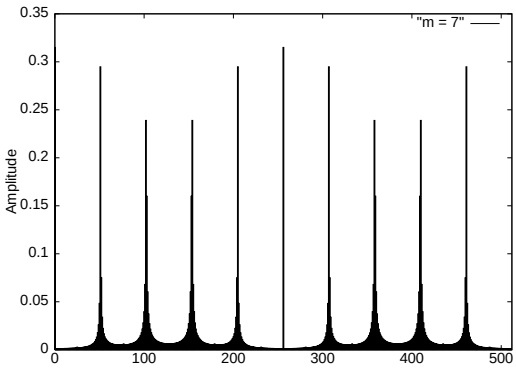
2.2 近似量子フーリエ変換のシミュレーション結果

図 3 に，近似量子フーリエ変換のシミュレーション結果を示す．近似の次数 m というのは，量子回路で書いた際に m 以上離れたところの相関をとる部分を除くということで，図 2 の例は次数 2 の例である．特に $m = 1$ のときは，有名な Walsh-Hadamard 変換となる．

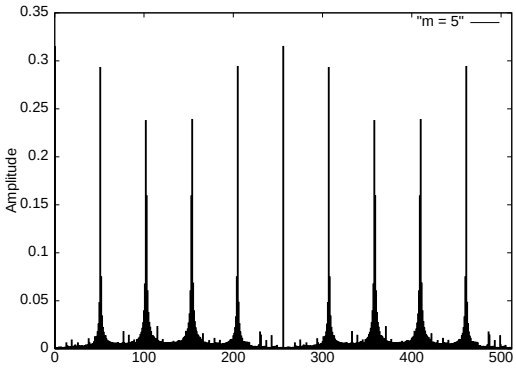
この図から，次数が小さくなる (近似の度合いが高くなる) に従って，どのくらいの精度で近似が実現されているかを定量的に見ることができる．上にも述べたように近似量子フーリエ変換したあとの結果は近似になっているが，それは単に観測した際に所望のデータが測定される成功確率に影響するのみであるため，この場合では $m = 3$ くらいまでの近似でも十分よい近似になっているといえ，それがシミュレーションを通して定量的に観察できる．



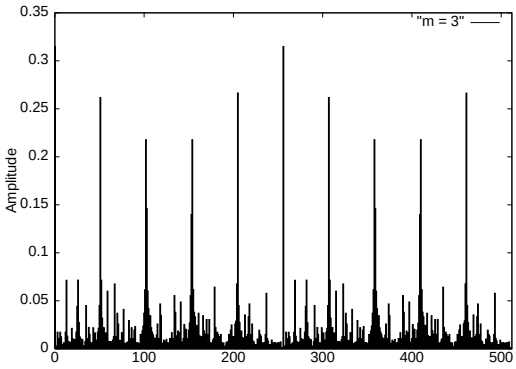
(a) $m = 9$ (QFT)



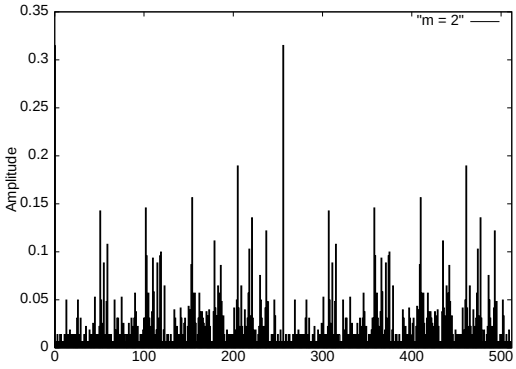
(b) $m = 7$



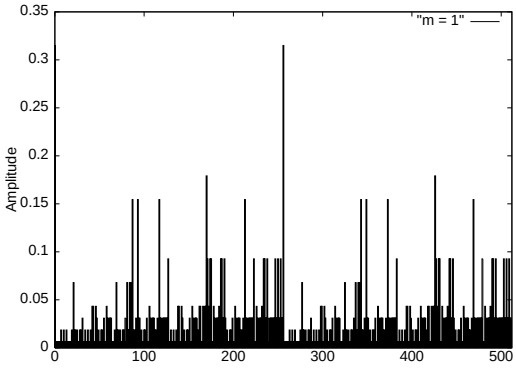
(c) $m = 5$



(d) $m = 3$



(e) $m = 2$



(f) $m = 1$ (Hadamard Transform)

図 3: (近似) 量子フーリエ変換結果. 近似次数 m の近似量子フーリエ変換を，Shor のアルゴリズムで現れる周期性を有する状態の 9 量子ビットに対して適用した結果. 横軸が $2^9 = 512$ の量子ビットを，縦軸が各々の振幅を表わす. m が小さくなった際の近似がひどくなる度合いが見て取れる.

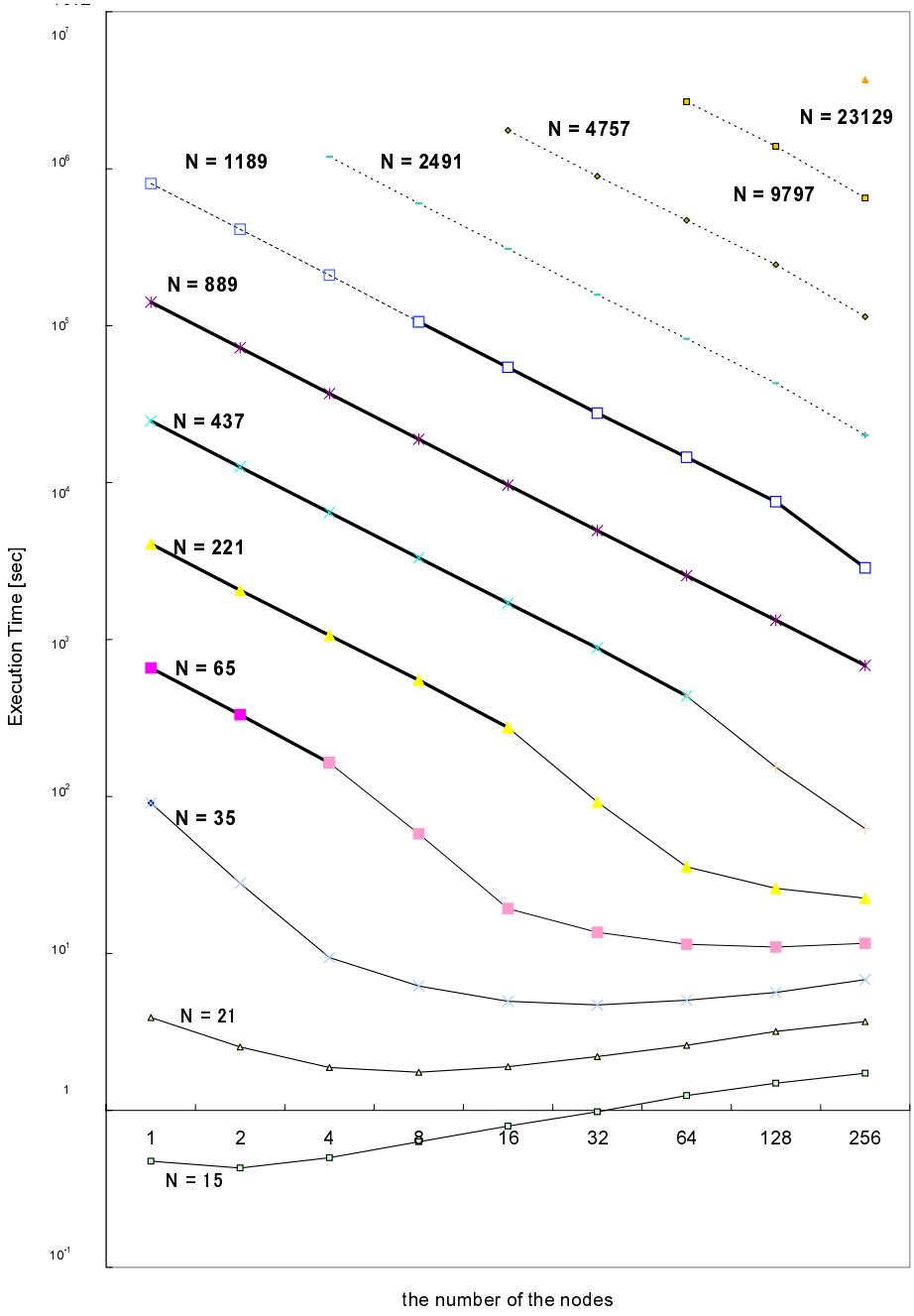


図 4: 素数 2 つの積 N を Shor の量子アルゴリズムで素因数分解するのを SCore-III でシミュレートした結果．横軸がプロセッサ数，縦軸が実行時間を示す．実線で結んでいるのが実際の結果で，その性質のよさから外挿できかつ SCore-III で原理的に解ける場合を点線で描いている

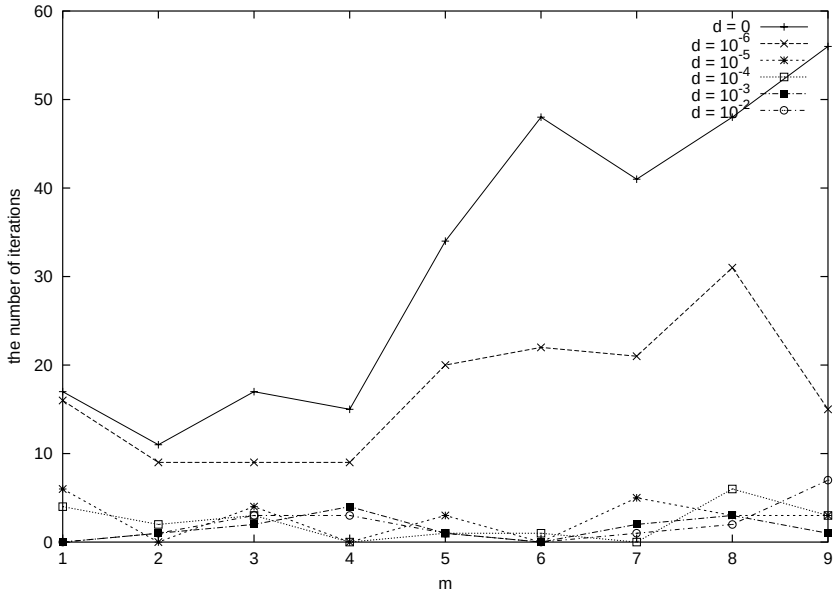


図 5: 近似量子フーリエ変換をデコヒーレンスがある状態で計算した場合のシミュレーション結果．デコヒーレンス確率 d が 10^{-5} 以下である場合，近似次数にほぼ関わらないくらいで Shor の素因数分解量子アルゴリズムがうまく動いている．横軸は近似次数で，縦軸は Shor のアルゴルズムで成功するまでの反復回数，図中の折れ線で各デコヒーレンス確率に対する結果を示している．

2.3 Shor のアルゴリズムの並列実装によるシミュレーションとその結果

現在のコンピュータで量子計算をシミュレートする場合，量子ビットに対する指数的なメモリが必要であることと，あと計算時間が伴って指数的な時間がかかるという問題点がある．現在のコンピュータで超大メモリ・超高速な環境を提供するのが分散並列コンピュータであり，本研究ではその成果をフルに活用して初めて可能になるサイズまでシミュレーションを行った．

図 4 に，SCore-III (PC クラスタシステム) で並列シミュレーションに要した時間を図示する．量子計算はテンソル積空間での定型的な線形計算としてシミュレーションできるので，その点を加味したシステムを開発することにより，十分大きな台数がある場合には理想的な並列効果が上がっていることが確認できる．

これによって，Shor のアルゴリズムのようにある程度複雑な量子アルゴリズムであっても，既存の技術で三十数量子ビットまでのシミュレーションが可能であることを示せた．

シミュレーションでは，量子アルゴリズムに特有であるデコヒーレンスエラー (各量子ビットについて，パウリのスピン行列で単位行列以外の 3 つが $d/3$ の確率で起るという最も基本的なモデル), 作エラー (量子操作した際に位相に正規分布でエラーが入るモデル) の両方のエラーを加えてシミュレーションを行った．

本報告では，Shor のアルゴリズムで AQFT を用いてデコヒーレンスエラーが存在する状態で調べた結果を示す．ここでは Shor のアルゴリズム全体が成功するまでの反復回数 (縦軸) についてのデータのみ図 5 に示す．

この図より，近似量子フーリエ変換の次数に比べ，デコヒーレンス確率 d の方が成功確率に強く影響をしており，d が 10^{-5} であることが必要であるのに対して，近似量子フーリエ変換の次数は小さくても (近似度をあげても) よいことがわかる．このような性質を理論的に解析することが期待され，またシミュレーションによってこそ，現状ではそのような性質を明らかにできる．