

QR 分解に関する高性能計算技術の研究

深谷 猛（北海道大学・情報基盤センター）

概要

本研究課題では、基本的な行列計算の一つである QR 分解に対して、様々な観点から研究開発に取り組む。課題参加者は、これまで独立して QR 分解に関連する研究を行った実績があり、本課題では、そのような研究者同士が協力して課題の解決に取り組むことで、QR 分解とその関連技術に関して、新しい技術や知見を創出することを目指す。継続課題の 2 年目として実施した 2024 年度は、前年度に開発したアルゴリズムの改良等の成果を得ることができた。同時に、QR 分解の新たな応用先を検討するとともに、課題を実施する上で蓄積されているプログラム等をベンチマーク向けに整備する取り組みも行った。2025 年度も継続課題として実施しており、引き続き、「QR 分解を共通のキーワードにした研究者の協調」を掲げて、QR 分解と関連技術の研究開発を進め、最終的に様々な応用分野のアプリケーションの高性能化に貢献することを目指す。

1 共同研究に関する情報

1.1 共同研究を実施した拠点名

- 北海道大学 情報基盤センター
- 東北大学 サイバーサイエンスセンター
- 東京大学 情報基盤センター
- 名古屋大学 情報基盤センター
- 京都大学 学術情報メディアセンター
- 九州大学 情報基盤研究開発センター

1.2 課題分野

- 大規模計算科学課題分野

1.3 参加研究者の役割分担

- 深谷 猛（北海道大学）：課題代表、全体統括、コレスキー QR アルゴリズム関連の研究開発
- 鈴木 智博（山梨大学）：課題副代表、統括補佐、タイルアルゴリズム関連の研究開発
- 大島 聡史（九州大学）：BLR 行列の QR

分解関連の研究開発、GPU 実装に関する助言

- 伊田 明弘（海洋研究開発機構）：BLR 行列の QR 分解関連の研究開発
- 岩下 武史（京都大学）：QR 分解の応用（線形ソルバー関連）
- 佐竹 祐樹（北海道大学）：QR 分解の応用（行列方程式関連）
- 阿部 龍仁（北海道大学・大学院生）：Randomized アルゴリズムに関する性能評価

2 研究の目的と意義

行列を都合のよい行列の積に分解する計算（行列分解）は、数値線形代数分野の基本技術の一つであり、様々なアプリケーションにおいて利用されるとともに、数値線形代数アルゴリズムを構成する部品としての役割を持つ。そのため、行列分解計算の高性能化（高速化）は重

要な課題であり、ハードウェアの特徴を考慮した上で、アルゴリズムから実装方法まで多岐にわたる研究開発が行われている。

本研究課題では、主要な行列分解計算の一つである、行列の QR 分解を扱う。QR 分解は与えられた行列を直交行列と上三角行列の積に分解する計算であり、最小二乗問題が代表的な応用列として知られている。また、行列の固有値計算や特異値計算とも密接な関わりを持っている。加えて、ベクトルの直交化（直交基底の生成）が QR 分解と等価であるため、数値安定性を向上させる目的等で、（ブロック版の）クリロフ部分空間法などにおいても需要がある。したがって、QR 分解の性能を向上させることで、様々な科学技術計算の効率化に貢献できる。

現在、QR 分解に対して異なる特徴を持った多様な数値計算アルゴリズムが存在する。一方、QR 分解の計算が行われる環境も、マルチコア CPU、GPU、分散並列システム（CPU と GPU が混在するヘテロな環境含む）と多種多様である。更に、計算対象の行列も縦長行列から正方行列まで様々な形状があり、加えて、最近では BLR (Block Low Rank) 行列の QR 分解のような問題設定も登場している。

このような背景を踏めた上で、本研究課題では、QR 分解とその関連技術に関して、異なる知識や研究実績を有する研究者を集め、協力して研究開発に取り組む。具体的には、参加する各研究者がこれまでに個別に研究開発をおこなってきた技術や知識を土台として、それらを相互に組み合わせることで、QR 分解とその関連技術の高性能化に資する新しい技術や知見を創出することが大局的な目的である。

本課題は 2023 年度から実施しており、2024 年度は 2 年目となる。2024 年度は、2023 年度から実施している 4 つの項目（後述の項目 1 から 4）については、アルゴリズムの改良や詳細

な性能評価の実施などを主たる目標として設定する。また、2024 年度から新しく実施する項目として、QR 分解の応用に関する項目と、課題で得られた成果物をベンチマークとして整備する項目を新たに加える（項目 5 と 6）。

3 当拠点公募型研究として実施した意義

JHPCN 課題では、特徴の異なる多種多様な計算機環境を使用可能であり、QR 分解に代表される基本的な線形計算技術の研究開発に適している。例えば、統一的な視点で、各計算機上で、特徴の異なる複数のアルゴリズム（や実装）を評価・分析することで、アルゴリズム開発者とアルゴリズム利用者の双方に有益な知見を得ることができる。また、開発したアルゴリズムをライブラリとして整備・公開することで、各センターで実行されているアプリケーションの高性能化に貢献できる。更に、JHPCN では、計算科学、データ科学、機械学習などの幅広い応用分野の研究課題が毎年実施されており、その課題の関係者とシンポジウム等で交流することができ、基盤技術を扱う本課題にとって大きな意義・利点がある。

4 前年度までに得られた研究成果の概要

課題実施 1 年目である、2023 年度に得られた主な成果は以下の通りである。

- 縦長行列の列ピボット付き QR 分解 (QRCP) に対して、コレスキー QR 型アルゴリズムを開発した。マルチコア CPU および分散並列環境で性能評価を実施し、既存手法に対して十分な優位性があることを確認した。得られた成果は IPDPS2024 の論文等で発表済みである。

- 非縦長行列の QR 分解に対して、反復型のコレスキー QR アルゴリズムとブロック版の Gram-Schmidt アルゴリズム (BCGS2) を組み合わせた手法を考案し、一次元のデータ分散を採用した分散並列版のプログラムを開発した。北大 Grand Chariot や東大 BDEC (Odyssey) を用いた性能評価を実施し、提案手法の有効性を検証した。得られた成果は HPC 研究会で発表済みである。
- 最新のマルチコア CPU 環境および分散並列環境において、開発したタイル版の QR 分解プログラムの動作検証を実施した。また、構造が QR 分解よりシンプルなコレスキー分解のアルゴリズムを対象に、分散並列環境でのタイル版アルゴリズムの性能評価等を実施し、QR 分解のタイル並列版プログラムの研究開発に有益となる知見を得た。なお、コレスキー分解に関する上記の成果については、HPC Asia 2024 で発表済みである。
- GPU 環境向けの BLR 行列の QR 分解アルゴリズムのプログラムの整理および分析を進めた。また、MIG の利用など GPU の効率的な利用方法など、関連する GPU 利用技術についても、他の JHPCN 課題などと連携して研究を進めた。得られた成果の一部は HPC 研究会で発表済みである。

5 今年度の研究成果の詳細

研究計画に記載した各項目に関する成果は以下の通りである。

5.1 縦長行列の列ピボット付き QR 分解に対するコレスキー QR 型アルゴリズムの開発

縦長行列の列ピボット付き QR 分解に関して、2023 年度に開発したコレスキー QR 型アルゴリズムの実装方法の改良に取り組んだ。具体的には、まず、アルゴリズム内部で用いる、グラム行列のコレスキー分解相当の計算を行うカーネルに関して、途中で計算を打ち切ることを考慮し、無駄になる計算量を削減する形に計算順序を変更した (改良版 1)。次に、グラム行列を計算するカーネル (行列積相当) に関して、行列積の計算をブロック化することで数学的にゼロとなる要素の計算を省略できるように変更した (改良版 2)。以上の 2 点の改良により、理論上は計算量を削減することができたが、一方で、計算の細分化や計算パターンの変更に伴う実行効率の低下の可能性があるため、計算時間の意味での利点に関しては、実際にプログラムを実行して評価することが必要である。図 1 に検証結果の一例を示す。図から分かるように、対象の行列の列数 (n) が大きい場合には、上記の 2 つの改良の有効性が確認できた。逆に、列数が小さい場合には逆効果となる場合があることも分かった。この結果を踏まえて、今後は、入力された行列のサイズに応じた実装の自動選択機能の実現などを念頭に置いて、更なる改良を進める。

上記の改良に加えて、開発したコレスキー QR 型アルゴリズムの有効性をより明確にするため、これまでに比較を行っていなかった、いくつかの既存手法との性能評価を実施した。具体的には、縦長行列に対する通常の QR 分解を行った後、得られた小規模の上三角行列に対して QRCP を行うことで、所望の結果を得る手法である。図 2 は検証結果の一例である。図のグラフより、問題サイズに応じた開発手法の

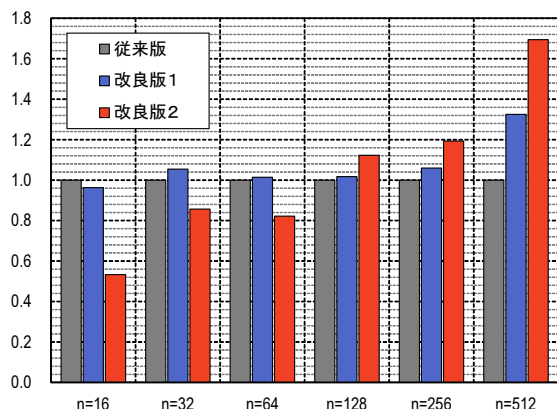


図1 縦長行列の QRCP に対するコレスキー QR 型アルゴリズムの改良版の性能検証結果の例：従来版に対する速度向上を評価（環境：北大 Grand Chariot 256 ノード、2 プロセス/ノード、20 スレッド/プロセス）

有効性がより明確になった。更に、低ランク近似等の応用を想定し、QR 分解の計算結果の一部のみを求める状況を想定した性能検証も実施した。具体的には、行列 Q の先頭から指定された範囲のみを求める状況について、各アルゴリズムを実行し、計算時間を比較した。図 3 は結果の一例である。我々が開発した手法は、計算を途中で打ち切ることで、所望の部分の結果を得ることが可能であるが、一方で、いくつかの既存手法は、行列全体に対する計算を避けることができない。そのため、図から明らかなように、このような状況においては我々の手法の有効性がより大きくなることを確認できた。

以上の通り、2023 年度に開発したアルゴリズムの改良や有効性のより詳しい検証を 2024 年度に実施することができた。

5.2 非縦長行列の QR 分解に対するコレスキー QR 型アルゴリズムの適用

2023 年度に開発したアルゴリズムに対して、過去に実施したハウスホルダー QR 型アルゴリズムのブロック分割方法の最適化の事例を

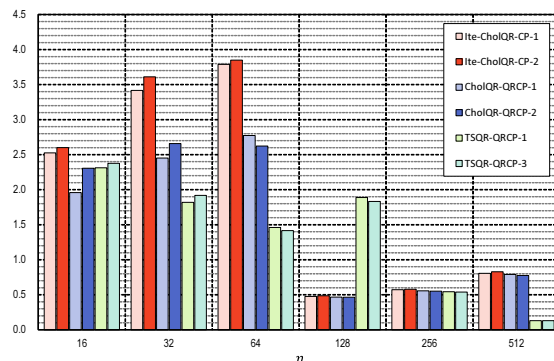


図2 縦長行列の QRCP に対する様々な既存手法と開発したコレスキー QR 型アルゴリズムの性能比較結果の例：ハウスホルダー QR アルゴリズムに対する速度向上を評価（環境：東大 Wisteria/BDEC-01(Oddasy)、4096 ノード、4 プロセス/ノード、12 スレッド/プロセス）

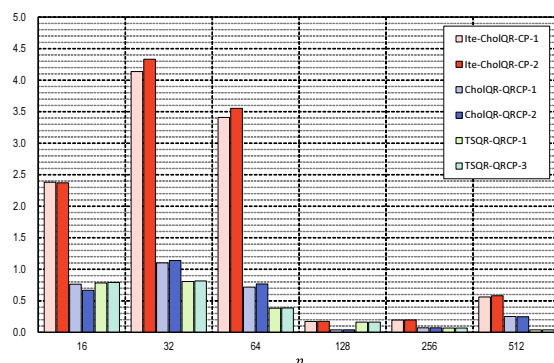


図3 縦長行列の QRCP に対する様々な既存手法と開発したコレスキー QR 型アルゴリズムの性能比較結果の例（低ランク近似を想定し、一部の結果のみを計算する場合）：ハウスホルダー QR アルゴリズムに対する速度向上を評価（環境：東大 Wisteria/BDEC-01(Oddasy)、4096 ノード、4 プロセス/ノード、12 スレッド/プロセス）

参考に、ブロック幅と性能の関係を予測するモデルの構築と、それを応用したブロック分割幅の決定方法の実現に向けた検討を進めた。具体的には、アルゴリズムを構成する BLAS 相当

のカーネルのベンチマークを実施し、その結果を用いたアルゴリズムの内部カーネルの各実行時間の予測値を積算することで、各ブロック幅における実行時間の予測を行うアプローチである。実際に BLAS カーネルのベンチマークを実施したところ、実行毎の時間の違い等が確認され、2024 年度は、その対応の検討などの段階までとなった。

また、アルゴリズム内部で用いるコレスキー QR 型アルゴリズムの反復回数に関して、最近の論文の結果より、元々、我々が想定していた回数から削減をすることができる可能性があることが分かった。そこで、既存の論文の理論解析の結果を参考にしながら、我々が想定している状況における反復回数の削減の可否について、検討を進めた。なお、現状、理論的に削減可能、という結果を得るには至っていない。

5.3 最新の計算機環境におけるタイル QR 型アルゴリズムの性能評価

2023 年度に実施した性能評価結果を踏まえて、東北大の AOBA-A システムを主たる対象として、性能評価の継続と改良版の分散並列版タイル QR アルゴリズムの実装に取り組んだ。また、2023 年度の実施分を含めた性能データを分析し、実装方法の改良やプログラム中のパラメータのチューニング手法の検討を進めた。図 4 はその一例であり、ブロックサイズとプログラムの性能の関係に関する調査結果である。この図のような結果を踏まえて、キャッシュメモリとの関係性などに着目し、パラメータの（自動）チューニング手法の開発に向けた研究を実施した。

5.4 BLR 行列の QR 分解アルゴリズムの改良と GPU 環境での性能評価

関連する別の課題と協力した上で、GPU 環境向けに開発中の BLR 行列の QR 分解プログラムの性能評価と分析を進めた。プロファイ

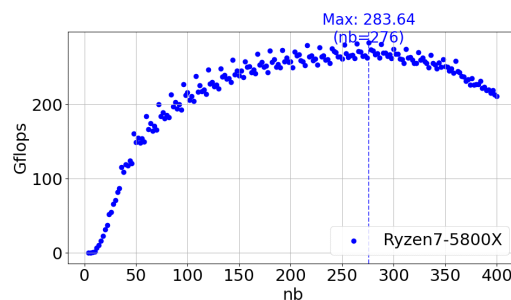


図 4 タイル QR アルゴリズムのブロックサイズと性能の関係の調査結果例

ラを用いた分析の結果などから GPU 環境で実行する場合のボトルネックを調査し、その部分の改良に関する検討を行った。例えば、既存の実装では行列の低ランク化のために特異値分解ルーチンを利用している箇所（図 5）に関して、理論上は演算コストが少ない RRQR ルーチンで代用するアプローチを試した。一方で、GPU 環境では、CPU 環境よりも利用可能な行列計算ルーチンの数が少ない、といった障壁もあり、理論上は有望でも、（少なくとも現状では）実用上難しい、という場合があることも分かった。加えて、GPU の計算リソースをより有効に活用するための手法（例：MIG を使うかどうか）も検討の余地が大きく、引き続き、関連する課題と協調して、多角的に有効なアプローチの検討と検証を進める必要があることが確認された。

5.5 QR 分解を必要とする線形計算アルゴリズムへの応用

縦長行列の QRCP 向けのコレスキー QR 型アルゴリズムの応用先として、大規模疎行列を係数とする連立方程式に対する混合精度型ソルバーの研究に取り組んだ。特に、前処理を伴う混合精度型ソルバーの数値的な振る舞いを検証し、一定の範囲の問題に関しては、従来の（倍

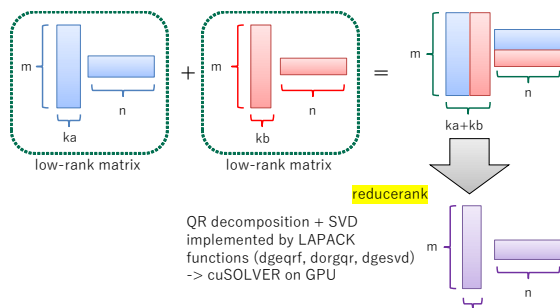


図5 BLR 行列の QR 分解アルゴリズム中で必要となる行列の低ランク近似の処理

精度のみを用いる）ソルバーと収束性が同程度であることを確認した。この結果を踏まえて、実際に、コレスキー QR 型の QRCP を用いた混合精度型ソルバーの設計に着手した。

また、行列方程式に対する低ランク近似を用いた求解法に対して検討を行い、直交化（QR 分解）を導入することでアルゴリズムを改良できる可能性があることを見出した。この結果に基づいて、既存のアルゴリズムを再構成し、理論的には、演算量や使用メモリ量の観点で優位性が期待できることを確認している。今後、まず、プロトタイプ版を実装して数値的な振る舞いを検証し、更に、実用を想定したサイズの問題に対して、実行時間の検証等を行う計画である。

5.6 QR 分解に関するライブラリの整備とベンチマークの実施

本課題において、2023 年度から実施している、縦長行列に対する QR 分解を計算する各アルゴリズムのプログラムコードをベンチマークとして公開することを念頭に整備した。現状、プログラムコードは Fortran 90 を用いて開発しているが、一方で、今後に向けて、適当な言語（やフレームワーク等）の調査や検討も進めている。また、近年、応用数学の分野で、乱択アルゴリズム（Randomized Algorithm）の研究が

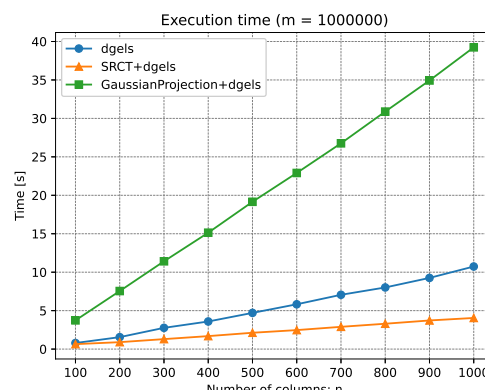


図6 乱択アルゴリズムのプロトタイプ版に関する性能評価結果例：素朴な乱択アルゴリズム 2 種類と既存（古典）アルゴリズムの実行時間を評価：（環境：北大 Grand Chariot、1 ノード、40 スレッド）

活発であり、縦長行列に対する乱択アルゴリズムのプロトタイプ版の実装にも着手した。図6は乱択アルゴリズムのプロトタイプ版に対する性能評価結果の一例である。2 種類の乱択アルゴリズム（SRCT と用いた手法と Gaussian を用いた手法）と既存（古典）の手法（dgels）の実行時間を問題サイズを変えながら測定して結果を示している。図から分かるように、SRCT を用いた手法は非常に高速であり、検討する価値が十分にあることが確認できた。一方で、計算精度に関する検証（図7が一例）も必須であり、GPU 向けの実装などをともに、今後、研究開発を継続する予定である。

6 今年度の進捗状況と今後の展望

課題全体としては、アルゴリズムの改良など着実な成果をいくつかの実施項目で得ることができた一方で、配分された計算資源の有効活用の点と、成果発表（特に論文等）の点で反省点が残る年度であった。計算資源の活用が限定的となってしまった要因の一つとして、主担当の

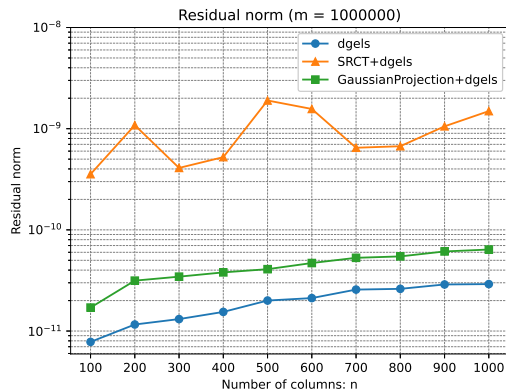


図7 乱択アルゴリズムのプロトタイプ版に関する計算精度の評価結果例：素朴な乱択アルゴリズム2種類と既存（古典）アルゴリズムの精度（計算結果の相対残差ノルム）を評価

教員がプログラム開発や性能評価を実施できる時間が限られてしまい、一方で、指導している学生等の協力が当初の想定よりも難しかったことが挙げられる。理論的な部分での進展は十分に得られたと考えるが、様々な計算資源を活用できる点が JHPCN 課題として実施している大きな意義の一つである以上、翌年度の改善を検討したい。

2023 年度から継続している項目（1 から 4）に関しては、項目間で差はあるが、それぞれ一定の進捗が得られたと考える。例えば、項目 1 については、論文発表には至っていないが、本稿で紹介したように、アルゴリズムの性能改良を実現できた。一方、2024 年度に新しく取り組んだ項目（5 と 6）については、どちらも一定の成果が得られており、翌年度に配分された計算資源を用いて性能評価などを実施するための一定の準備ができたと考える。

また、本課題の趣旨の一つである「QR 分解を共通のキーワードにした研究者の協調」という点では、目に見える成果として評価するのは難しいが、チャットなどにおける日常の議論の

活発化などの点で成果が出ていると考える。今後、共著論文等の目に見える形での成果につなげることが大きな目標となる。

2025 年度も継続課題として実施するが、項目 1 から 4 に関しては、2024 年度に実施できなかった点を適宜実施し、追加の実施内容と併せて、論文等の成果として発表することが求められる。また、アルゴリズムの開発・改良を伴い、多岐にわたる各項目の全てを計画通りに実施するのは簡単ではないので、計画を必要に応じて柔軟に修正し、利用可能な計算資源を活用するように努めたい。