複素非対称行列向け固有値解法の CSX600 による高速化

宮田 考史[1]，山本 有作[2]，中村 佳正[3]

[1,2]名古屋大学大学院工学研究科計算理工学専攻 [3]京都大学大学院情報学研究科数理工学専攻

[1] miyata@na.cse.nagoya-u.ac.jp [2] yamamoto@na.cse.nagoya-u.ac.jp [3] ynaka@amp.i.kyoto-u.ac.jp

概要 : 本稿では，複素非対称行列に対するヘッセンベルグ QR 法を ClearSpeed 社の浮動小数点コプロセッサ CSX600 を用いて高速化した結果について報告する．Braman らにより提案された small-bulge マルチシフト QR 法は，計算の大部分を行列乗算の形で実行できるため，高性能計算機に適している．しかし，CSX600 の性能を引き出すためにシフト数を増やすと，行列乗算以外の部分の実行時間が無視できなくなる．そこで，本稿では small-bulge マルチシフト QR 法を修正することで，従来は行列乗算が利用できなかった部分に対して，行列乗算の利用を可能にした．これにより，12,000×12,000 の複素ヘッセンベルグ行列の固有値とシューア標準形を求める場合，CSX600 を使うことで 3.2GHz Xeon に比べ 3.8 倍の高速化を達成できた．

1 はじめに

複素非対称密行列の固有値問題は，様々な応用分野で重要であり，近年では 1 万元を超える大型の問題も解かれるようになっている．この問題を解く標準的なアルゴリズムは，ユニタリ変換によるヘッセンベルグ行列への変形，ヘッセンベルグ QR 法による固有値とシューア標準形の計算，固有ベクトル計算から成り立つ．多くの場合，QR 法が計算時間の大部分を占めるため，この部分の高速化が重要である．

ヘッセンベルグ QR 法には様々な変種があるが，Braman らの small-bulge マルチシフト QR 法[1]（以下，マルチシフト QR 法）が有力である．このアルゴリズムは，複数のシフトを導入することで QR 法の基本演算であるバルジ追跡演算を複数個まとめて行う．さらに，計算の大部分を行列乗算で実行可能なため，高性能計算機に適している．

近年，1 チップに多数の浮動小数点演算器を集積した専用プロセッサが開発され，大規模計算を高速に行う手段として注目を集めている．例えば CSX600[2]は 1 チップに 96 個の演算コアを集積し，最大 48GFLOPS の性能を持つ．

本稿では行列乗算を高速に実行可能な CSX600 を用いて，マルチシフト QR 法を高速化する．アルゴリズムの中で，行列乗算のみを CSX600 に担当させ, それ以外を CPU で実行することにより，ある程度の高速化は可能である．しかし，CSX600 の性能を引き出すには，シフト数を大きくする必要がある．一方，シフト数の増加に伴い，行列乗算以外の部分の割合が増え，ボトルネックとなる．

この問題を解決するため，我々はマルチシフト QR 法のアルゴリズムを修正し，従来は行列乗算が利用できなかった部分に対しても，行列乗算の利用を可能にする．

2 マルチシフト QR 法とその改良

2.1 再帰型アルゴリズムへの変換

マルチシフト QR 法のアルゴリズムはバルジ追跡と呼ばれる逐次的な演算の反復である．シフト数を m とすると，マルチシフト QR 法はハウスホルダー変換を用いて$(m/2)$個のバルジを，それぞれ行列の左上隅から右下まで追跡することにより，ダブルシフト QR 法の演算を$(m/2)$個まとめて行う．バルジをそれぞれ k 行ずつ追跡することを考えると, 影響を受ける領域は, 図 1 における$(m/2)\times 3+k$ 行，$(m/2)\times 3+k$ 列の領域である．これを対角ブロック(a)，非対角行ブロック(b)と非対角列ブロック(c)の更新処理に分割する．

バルジ追跡によるハウスホルダー変換は，対角ブロック内での演算のみから定まるから，行列全体の更新処理を次の 3 つに分けて実行可能である．

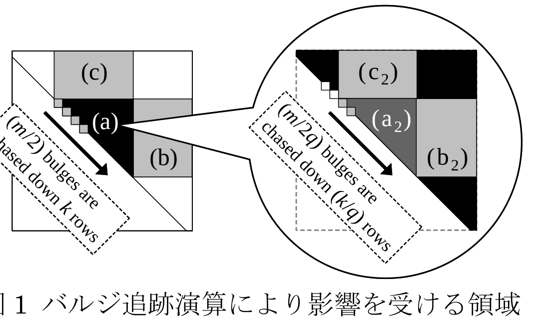

図 1 バルジ追跡演算により影響を受ける領域

(a) 対角ブロック内部で，$(m/2)$個のバルジをそれぞれ k 行追跡する．また，そこで使ったハウスホルダー変換を，$((m/2)\times 3+k)\times((m/2)\times 3+k)$のユニタリ行列 U に累積する．

(b) 非対角行ブロックに左から U をかける．

(c) 非対角列ブロックに右からU^*をかける．

更新処理(b)，(c)は，正方行列Uと非対角ブロックの行列乗算で実行可能である．Bramanらは，演算量最小化のためには$k=(m/2)\times 3$ とすることが最適であることを示している．このようにkを決め，シフト数mを増やすと，行列乗算の性能が向上できる．一方，対角ブロックの演算量が増大するが，この部分は行列乗算が利用できないため，ボトルネックになりやすい．そこで，対角ブロックの更新をq分割し，$(m/2q)$個のバルジをそれぞれk/q行追跡する．これにより，対角ブロック内部の演算（図１の(a)）を，さらに対角ブロック（図１の(a_2)）と非対角ブロック（図１の(b_2)，(c_2)）の更新処理に分割する．非対角ブロックの更新は行列乗算で実行可能である．

このように対角ブロック内部の演算を再帰的に分割することで，行列乗算の利用を可能にする．

2.2　ブロック化

マルチシフト QR 法の反復が進むにつれて行列の副対角要素は減少し，0 に収束する．この時点で右下隅の小さな $M\times M$ 小行列を分離し，これに対してダブルシフト QR 法を適用して固有値とシューア標準形を求める．ここで M は，通常 m に近い整数である．このとき，ダブルシフト QR 法のバルジ追跡で使うハウスホルダー変換は，元の行列の最後の M 列全体に作用させる必要がある．これらの処理をまとめて「分離した小行列の固有値計算」と呼ぶ．これが終了したら，残りの行列に対して再びマルチシフト QR 法を適用する．

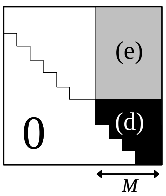

図２　分離した小行列の固有値計算で影響を受ける領域

シフト数mを増やすと分離した小行列の固有値計算が増大し，ボトルネックになる可能性がある．そこで，この部分に対しても更新処理を分割する．最初に対角ブロック（図２の(d)）にダブルシフトQR法を適用して固有値とシューア標準形を求め，そのとき使ったすべてのハウスホルダー変換を$M\times M$のユニタリ行列Pに累積する．最後に非対角ブロック（図２の(e)）に右からP^*をかける．これにより，非対角ブロックへの作用を行列乗算で行うことができ，性能が向上する．さらに従来よりも非対角ブロックの更新部分で演算量を削減することができる．

3　性能評価

本稿で改良したマルチシフト QR 法のプログラムを FORTRAN で作成し，従来法との性能比較を行った．計算機環境は Xeon 3.2GHz，メモリ 8GB，Intel Fortran Compiler Ver. 9.1 である．また，CSX600 チップが 2 個搭載された ClearSpeed Advance ボードと CSX600 用の BLAS を提供する CSXL Library Ver. 2.2 を用いた．プログラムの中で，行列乗算のみを CSX600 に担当させ，それ以外は CPU で実行した．テスト問題は実部と虚部が[0,1]の一様乱数に従う $n\times n$ 複素非対称行列である．ハウスホルダー法でヘッセンベルグ化したものを入力とし，すべての固有値とシューア標準形を求めた．結果を図 3 に示す．

図 3 より，提案法は従来法に比べて 1.4 倍高速である．非対角ブロックの更新演算は，従来法，提案法ともに行列乗算である．異なるのは，対角ブロックの更新と小行列の固有値計算部分である．それぞれ再帰型アルゴリズムへの変換とブロック化を行ったことにより，提案法では計算時間を短縮することが出来た．また，n =12,000 のとき，提案法と CSX600 を組み合わせることで，Xeon 単体に比べて 3.8 倍の高速化を達成できた．

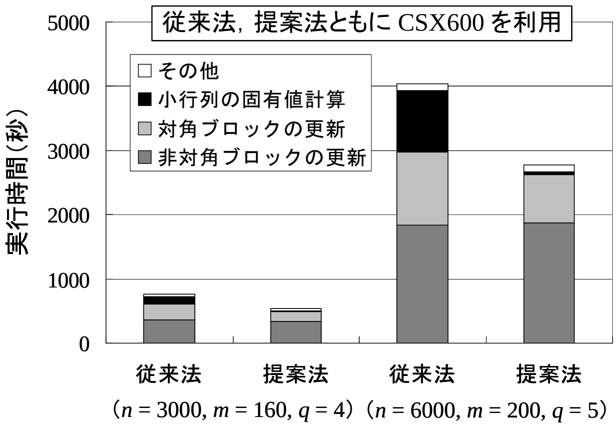

図３　アルゴリズムの効果

4　まとめ

本稿では，コプロセッサ向けのマルチシフト QR 法を提案した．提案法は CSX600 の性能を引き出し，Xeon 単体に比べ最大 3.8 倍の高速化を示した．

参考文献

[1] K. Braman, R. Byers and R. Mathias, The multishift QR algorithm part I: Maintaining well-focused shifts and level 3 performance, SIAM J. Matrix Anal. Appl., 23(4):929-947, 2002.

[2] http://www.cloarspeed.com/.