

## 高速特異値分解のためのライブラリ開発

高田 雅美<sup>†1,†2</sup>, 木村 欣司<sup>†3,†4</sup>  
 岩崎 雅史<sup>†1,†2</sup>, 中村 佳正<sup>†2,†1</sup>

高精度かつ高速に上 2 重対角行列を特異値分解するために、我々は、dLV (離散ロトカ・ボルテラ：discrete Lotka-Volterra) 系による新たな特異値分解ライブラリを開発している。既存ライブラリとしては、線形数値計算ライブラリ LAPACK における *DBDSQR* がある。*DBDSQR* は、QRs 法に基づいた特異値分解ライブラリであるが、計算量が多く、実行時間の面で大規模向きではない。また、いくつかの特異ベクトルのみを計算することも困難である。一方、dLV 系により定式化された *I-SVD* (Integrable-Singular Value Decomposition) 法は、計算量が抑えられる動作原理を持つ。本論文では、*I-SVD* 法の実装ライブラリ *DBDSLV* を開発し、実行時間と計算精度について、*DBDSQR* との比較数値実験を行う。精度を調べる際、真の特異値と特異ベクトルが判明している上 2 重対角行列が必要となる。そこで、Golub-Kahan-Lanczos 法によるテスト行列作成法を用いる。実験の結果、*DBDSLV* は、*DBDSQR* よりも誤差の少ない特異値と特異ベクトルを非常に短い実行時間で計算されることが確認された。計算された特異ベクトルの直交性も同程度であり、再直交化を行えば *DBDSQR* を上回ることも分かった。

## Implementation of Library for High Speed Singular Value Decomposition

MASAMI TAKATA,<sup>†1,†2</sup> KINJI KIMURA,<sup>†3,†4</sup> MASASHI IWASAKI<sup>†1,†2</sup>,  
 and YOSHIMASA NAKAMURA<sup>†2,†1</sup>.

To perform SVD (Singular Value Decomposition) of bidiagonal matrices with high accuracy and high-speed, we develop a library by using the dLV (discrete Lotka-Volterra) system. Today's standard routine for SVD is *DBDSQR* provided in LAPACK (Linear Algebra PACKage). Since the computation of SVD by *DBDSQR* is based on the QRs (QR with shift) algorithm, *DBDSQR* is slow in speed and is unsuitable for large scaled problems. It is also difficult to obtain only a few singular vectors by using *DBDSQR*. On the other hand, the *I-SVD* (Integrable-SVD) scheme based on the dLV system enables us to cut down the computational cost by separating the computation process of singular values and vectors. In this paper, for evaluation of computational time and accuracy, we implement the *I-SVD* scheme to a new routine named *DBDSLV* and compare it with *DBDSQR*. For a comparison of accuracy, we use a method for constructing a class of upper bidiagonal random test matrices having true singular values and vectors by means of the Golub-Kahan-Lanczos method. As experimental results, we confirmed that *DBDSLV* is faster and errors of singular values and vectors in *DBDSLV* are smaller than those in *DBDSQR*. Though the orthogonality of computed singular vectors in *DBDSLV* is in the same order as in *DBDSQR*, we found that *DBDSLV* has a better orthogonality through a reorthogonalization.

### 1. はじめに

データ検索<sup>19)</sup> や画像処理<sup>25)</sup> などにおいて、長方形行列を対角行列と 2 つの直交行列の積に分解する特異値分解が有効である。この対角行列の対角成分には特異値、直交行列の列には特異ベクトルがそれぞれ並ぶ。現在、与えられた密行列をあらかじめ上 2 重対角行列に変換して特異値分解を実行する方法が主流である<sup>2),8)</sup>。2 重対角行列の特異値分解には、QRs (原点シフト付き QR: QR with shift) 法<sup>2),3),7)-9),20),22)</sup>

†1 独立行政法人科学技術振興機構さきがけ  
 PRESTO, Japan Science and Technology Agency

†2 京都大学大学院情報学研究所  
 Graduate School of Informatics, Kyoto University

†3 独立行政法人科学技術振興機構 CREST  
 CREST, Japan Science and Technology Agency

†4 立教大学理学部  
 College of Science, Rikkyo University  
 現在、奈良女子大学大学院人間文化研究科  
 Presently with Graduate School of Humanity and Science, Nara Women's University  
 現在、独立行政法人科学技術振興機構発展研究  
 Presently with SORST, Japan Science and Technology Agency

がよく利用される。計算機の普及と性能の向上にもない、線形数値計算は、飛躍的な進歩をとげたが、その中であっても QRs 法は 20 世紀を代表するアルゴリズムの 1 つに数えられる。ところが、より多くのデータをより迅速に処理することが要求される今日では、QRs 法の計算量の多さは大きな問題となる。つまり、一般的な計算機において、QRs 法では大規模行列を現実的な時間で特異値分解することが困難である。データ検索や画像処理において、大きな特異値に対応する特異ベクトルのみを必要とすることがある。この際、必要な数個の特異ベクトルのみを計算することが困難な QRs 法では、メモリ空間や時間が過剰に消費されてしまう可能性がある。

我々は、可積分系による数値計算パッケージ LAPIS (Linear Algebra Package by Integrable Systems) を開発中である。これまでに、特異値を高速高精度に計算するための *mdLVs* (原点シフト付き修正離散ロトカ・ボルテラ：modified discrete Lotka-Volterra with shift) 法<sup>(12),(14),(15),(24)</sup> の有効な実装方法を提案し、計算速度と精度について示した<sup>(29),(30)</sup>。本論文では、*mdLVs* 法に特異ベクトル計算機能を追加した新たな特異値分解法 *I-SVD* (Integrable-Singular Value Decomposition)<sup>(12)-14),(24)</sup> の実装を行い、LAPACK<sup>(21)</sup> の代表的な 2 重対角行列の特異値分解ライブラリ *DBDSQR* との比較によって性能を評価する。*DBDSQR* の *DBD* は、引数が DOUBLE PRECISION 型で対象とする行列が BiDiagonal ということの意味し、*SQR* は QRs 法に基づく特異値分解ライブラリを示す。そこで、*I-SVD* 法の実装ライブラリも LAPACK にならって *DBDSLV* と名付ける。*SLV* は Shift 付き離散 Lotka-Volterra 系およびその変換を用いた特異値分解を指す。

特異値分解ライブラリの性能評価のために、計算された特異ベクトルの直交性を確認することが多い。もちろん、正しい特異ベクトルからのずれも判明すればよいが、行列サイズが大きい場合、正しい特異ベクトルを知ることは難しい。そこで本論文では、Golub-Kahan-Lanczos 法<sup>(8)</sup> を基とする数値実験のためのテスト行列作成法を用いる。この方法によって、特異値および特異ベクトルが正確にわかる上 2 重対角行列をランダムに作成できる。以降、計算された特異ベク

トルの直交性の度合い、真値との差のノルム (誤差の大きさ) をあわせて特異ベクトルの精度と呼ぶことにする。

2 章では、QRs 法と *I-SVD* 法について説明する。3 章において、Golub-Kahan-Lanczos 法を利用した上 2 重対角テスト行列の作成法について示す。4 章では、異なる *DBDSQR* と *DBDSLV* の計算精度および実行時間の比較・検証を行う。

## 2. 特異値分解法

### 2.1 QRs 法と I-SVD 法

$l \times m$  ( $l \geq m$ ) 長方形行列  $A$  に対して、適当な基本直交行列とその転置行列を左右から乗じても特異値は変化しない。同様の変形を多くとも  $(m-2)$  回繰り返せば、 $A$  と同じ特異値を持つ上 2 重対角行列  $B$  が得られる。これは、Householder 法<sup>(2),(8),(31)</sup> の 1 つの例で、 $A$  を効率良く特異値分解するためには欠かせない前処理である。 $B$  の特異値分解とは  $B = U\Sigma V^T$  なる分解をいう。ここで、 $\Sigma$  は対角行列、 $U = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m)$  は左直交行列、 $V = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$  は右直交行列、 $\Sigma$  の対角成分には  $B$  の特異値  $\sigma_i \geq 0$  ( $i: 1 \leq i \leq m$ ) が並び、 $\sigma_i$  に対応する特異ベクトルが  $\mathbf{u}_i$  および  $\mathbf{v}_i$  である。特異値分解のための信頼できる方法として、QRs 法<sup>(2),(3)</sup> と *I-SVD* 法<sup>(16)</sup> がある。

QRs 法において、左右直交行列は、直交行列の積を繰り返し計算することで与えられる。そのため、QRs 法を基盤とする *DBDSQR* は、左右直交行列の列として得られる特異ベクトルの直交性が必然的に高くなる。十分な改良が重ねられ非常に完成度が高い Golub-Kahan が提案<sup>(8),(9)</sup> した特異値分解法は、今もなお根強く支持されている。しかし、QRs 法は中規模向きで、計算された特異値の相対精度もあまり良くない<sup>(29),(30)</sup>。また、すべての特異ベクトルの計算は、ほぼ同時に終了するため、効率的にいくつかの特異ベクトルのみを計算することも困難である。

一方、*I-SVD* 法では、図 1 のように、特異値と特異ベクトルの計算プロセスが完全に 2 つに分離されている。図 1 では  $V, U$  の順に計算しているが、 $V$  と  $U$  の計算順序を逆にしてもよい。*I-SVD* 法の特異値計算部には、収束性と数値安定性が保証された *mdLVs* 法が採用されている。*mdLVs* 法では、QRs 法と比べ、高精度な特異値が高速に計算される<sup>(29),(30)</sup>。特異ベクトルは、*dLVv* (不等間隔差分ロトカ・ボルテラ：discrete Lotka-Volterra with variable step-size) 型変換を含む合成変換によって高精度に計算される<sup>(16)</sup>。この合成変換により、各特異値  $\sigma_i$  に対する特異ベクトルをそ

固有値分解法として、QRs 法のほかに  $MR^3$  (multiple Relatively Robust Representations) 法<sup>(4)</sup> がある。しかし、 $MR^3$  法は、収束性や数値安定性が不明瞭で信頼性が低い<sup>(32)</sup>。また、LAPACK (Linear Algebra PACKage)<sup>(21)</sup> には特異値分解用に  $MR^3$  法を実装したライブラリはない。ゆえに、本論文において、 $MR^3$  法は取り扱わない。

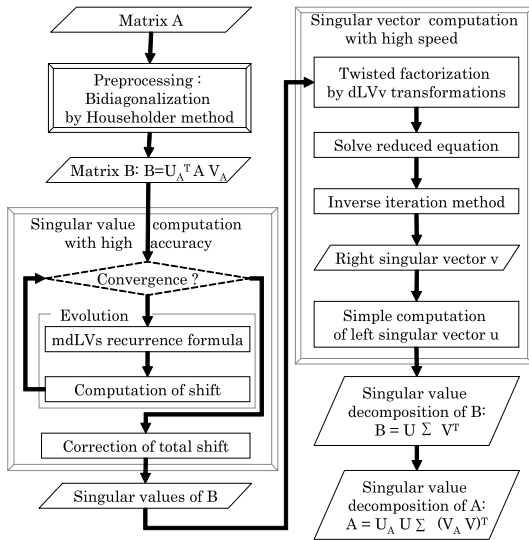


図1 特異値分解法  $I$ -SVD を実装したライブラリの流れ

Fig. 1 Computational flow of the library which realizes the singular value decomposition method  $I$ -SVD.

それぞれ独立に計算することができる。そのため、実用問題として数個の特異ベクトルのみを高速に計算したいという要求への対応は容易である。その結果として、実行時間の短縮やメモリ使用量の減少が見込まれる。

$mdLVs$  法は 2.2 節で、 $dLVv$  変換による特異ベクトル計算法については 2.3 節で説明する。

## 2.2 $mdLVs$ 法による特異値計算

上 2 重対角行列  $B$  の特異値のみを計算する方法として  $dqds$  (differential  $qd$  with shift) 法<sup>(6),(27),(28)</sup>がある。LAPACK には  $dqds$  法の実装ライブラリ  $DLASQ$  が公開されている。一般に、 $dqds$  法によって  $QRs$  法よりも短い実行時間でより高精度な特異値が計算されるが、有限桁精度計算のもとでの  $dqds$  法の収束証明は知られていない<sup>(32)</sup>。一方、 $mdLVs$  法では、特異値への収束性が保証されており<sup>(12),(14)</sup>、 $dqds$  法よりも高精度なアルゴリズムである<sup>(29)</sup>。実行時間は多くの場合  $dqds$  法の 2 ~ 3 倍程度であるが<sup>(29)</sup>、実行環境によっては  $dqds$  法よりも高速となる<sup>(30)</sup>。 $mdLVs$  法は  $QRs$  法よりつねに高速高精度である。

$mdLVs$  法の基本動作は以下に示す (i) ~ (iv) の繰返しである。対角成分が  $b_{2i-1}$ 、非対角成分が  $b_{2i}$  である  $m$  次上 2 重対角行列  $B$  の特異値  $\sigma_i$  を計算するには、 $x_k^{(0)} = b_k^2$  ( $k: 1 \leq k \leq 2m-1$ ) として (i) ~ (iv) を反復すればよい。この結果、発展回数  $n \rightarrow \infty$  で、 $x_{2i-1}^{(n)} \rightarrow \sigma_i^2 - \sum_{n=0}^{\infty} S^{(n)}$ 、 $x_{2i}^{(n)} \rightarrow 0$  と収束する。

$$(i) \quad x_k^{(n)} \mapsto y_k^{(n)}: \\ y_k^{(n)} = x_k^{(n)} / (1 + \delta^{(n)} y_{k-1}^{(n)}), \\ y_0^{(n)} \equiv 0, \quad \delta^{(n)} > 0$$

$$(ii) \quad y_k^{(n)} \mapsto z_k^{(n)}: \\ z_k^{(n)} = x_k^{(n)} (1 + \delta^{(n)} x_{k+1}^{(n)}), \quad x_{2m}^{(n)} \equiv 0$$

(iii) シフト量  $S^{(n)}$  の見積り:

Gersgorin 境界<sup>(11)</sup> or Johnson 境界<sup>(17)</sup> の計算

$$(iv) \quad z_k^{(n)} \mapsto x_k^{(n+1)}:$$

•  $S^{(n)}$  が有効ならば

$$\begin{cases} x_{2i-1}^{(n+1)} = z_{2i-1}^{(n)} + z_{2i-2}^{(n)} - x_{2i-2}^{(n+1)} - S^{(n)} \\ x_{2i}^{(n+1)} = z_{2i}^{(n)} z_{2i-1}^{(n)} / x_{2i-1}^{(n+1)} \end{cases}$$

• そうでなければ  $x_k^{(n+1)} = z_k^{(n)}$

$x_k^{(n)}$  と  $\delta^{(n)}$  は、発展回数  $n$  における要素 (変数)  $x_k$  と差分間隔をそれぞれ表す。 $\delta^{(n)}$  は、任意の正値をとることができる。

$mdLVs$  法の有効な実装ライブラリとして  $DLVS$  が構築されている<sup>(29)</sup>。 $DLVS$  では、行列を小さな次数の行列に分割する  $SPLIT$  や行列の次数を 1 つずつ下げる減次のほか、メモリ、キャッシュ、レジスタの有効活用のための様々な工夫が施されている<sup>(27),(29),(30)</sup>。

## 2.3 新しいツイスト分解による特異ベクトル計算法

$B$  の特異値  $\sigma_i$  に対して、実対称 3 重対角行列  $B^T B - \sigma_i^2 E$  を係数とする同次型連立 1 次方程式

$$(B^T B - \sigma_i^2 E) \mathbf{v}_i = 0 \quad (1)$$

の解  $\mathbf{v}_i = (v_i(1), v_i(2), \dots, v_i(m))^T$  は  $\sigma_i$  の右特異ベクトルを与える。なお、 $E$  は単位行列とする。正規化  $\mathbf{v}_i / \|\mathbf{v}_i\| \rightarrow \mathbf{v}_i$  を行えば、直交行列  $V$  の列ベクトルとなる。直交行列  $U$  は  $U = BV\Sigma^{-1}$  で計算され、特異値分解が完了する。ところが、計算機で得られる  $\sigma_i$  は、近似特異値である。また、式 (1) の解も次数  $m$  が大きくなると誤差なく計算することが困難となる。仮に、各  $\mathbf{v}_i$  が連立 1 次方程式の解として高精度に計算しても、 $\mathbf{v}_i^T \mathbf{v}_i \simeq 1$ 、 $\mathbf{v}_i^T \mathbf{v}_j \simeq 0$  ( $i \neq j$ ,  $j: 1 \leq j \leq m$ ) のような特異ベクトルの近似値として満たすべき直交性が優れているとは限らない。

そこで、 $\mathbf{v}_i$  が直交性の良い特異ベクトルとなるように、3 重対角対称行列のツイスト分解を活用した高速特異ベクトル計算法が考案された<sup>(5),(26)</sup>。この方法では、先に計算された近似特異値からツイスト分解によって近似特異ベクトルを計算する。さらに、逆反復法を 1 回行うだけで精度が向上した特異ベクトルが得られる。ツイスト分解法の核となる演算は、 $B^T B - \sigma_i^2 E$  の (複素型) Cholesky 分解

$$B^T B - \sigma_i^2 E = \begin{cases} (B^+)^T B^+, \\ (B^-)^T B^-, \end{cases} \quad (2)$$

$$B^+ \equiv \begin{pmatrix} \xi_1 \sqrt{b_1^+} & \xi_2 \sqrt{b_2^+} & & & \\ & \xi_3 \sqrt{b_3^+} & \ddots & & \\ & & \ddots & \xi_{2m-2} \sqrt{b_{2m-2}^+} & \\ 0 & & & \xi_{2m-1} \sqrt{b_{2m-1}^+} & \\ & & & & 0 \end{pmatrix}, \quad (3)$$

$$B^- \equiv \begin{pmatrix} \eta_1 \sqrt{b_1^-} & & & & 0 \\ \eta_2 \sqrt{b_2^-} & \eta_3 \sqrt{b_3^-} & & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_{2m-2} \sqrt{b_{2m-2}^-} & \eta_{2m-1} \sqrt{b_{2m-1}^-} & \\ & & & & 0 \end{pmatrix} \quad (4)$$

である。ただし、 $\xi_k = \pm 1$ ,  $\eta_k = \pm 1$  である。このとき、右特異ベクトル  $v_i$  の成分は、 $B^+$  および  $B^-$  の成分を用いて書き下される<sup>5),16),26)</sup>。

もちろん、近似特異値  $\sigma_i$  の精度が悪くは高精度な特異ベクトル  $v_i$  は期待できないが、*mdLVs* 法<sup>29),30)</sup> のような高精度手法によって特異値  $\sigma_i$  を計算すると、 $B^T B - \sigma_i^2 E$  が非正則行列に近づき、 $B^T B - \sigma_i^2 E$  の条件数が非常に大きくなる。このような場合、*qd* 型変換<sup>5),26)</sup> によって Cholesky 分解したのでは、桁落ちが発生して大きな誤差を含んだ値が得られることがある。これに対して、*I-SVD* では、Cholesky 分解のために以下のミウラ変換やその逆変換、*dLVv* 型変換を利用することで、数値安定かつ高精度に計算できる<sup>16)</sup>。具体的には、 $B^T B - \sigma_i^2 E = (B^+)^T B^+$  なる上 2 重対角行列  $B^+$  の成分  $\{\xi_{2i-1} \sqrt{b_{2i-1}^+}, \xi_{2i} \sqrt{b_{2i}^+}\}$  を計算するのに、以下の 3 つの変換を実行する<sup>16)</sup>。

- ミウラ変換： $\{b_{2i-1}, b_{2i}\} \mapsto \{x_{2i-1}, x_{2i}\}$   

$$\begin{cases} b_{2i-1} = \frac{1}{\delta} (1 + \delta x_{2i-2})(1 + \delta x_{2i-1}) \\ b_{2i} = \delta x_{2i-1} x_{2i} \end{cases}$$
- *stdLVv* (stationary *dLVv*) 変換：  
 $\{x_{2i-1}, x_{2i}\} \mapsto \{x_{2i-1}^+, x_{2i}^+\}$   

$$\begin{cases} x_{2i-1}^+ (1 + \delta^+ x_{2i-2}^+) = x_{2i-1} (1 + \delta x_{2i-2}) \\ x_{2i}^+ (1 + \delta^+ x_{2i-1}^+) = x_{2i} (1 + \delta x_{2i-1}) \end{cases}$$
- ミウラ逆変換： $\{x_{2i-1}^+, x_{2i}^+\} \mapsto \{b_{2i-1}^+, b_{2i}^+\}$   

$$\begin{cases} b_{2i-1}^+ = \frac{1}{\delta^+} (1 + \delta^+ x_{2i-2}^+) (1 + \delta^+ x_{2i-1}^+) \\ b_{2i}^+ = \delta^+ x_{2i-1}^+ x_{2i}^+ \end{cases}$$

ただし、 $b_{2i-1}$ ,  $b_{2i}$  はそれぞれ  $B$  の対角成分、非対角成分を 2 乗したものとする。パラメータ  $\delta$ ,  $\delta^+$  は  $1/\delta - 1/\delta^+ = \sigma_i^2$  を満たす範囲で自由に設定できる。 $\xi_k$  は、 $B^T B$  と  $(B^+)^T B^+$  の成分の符号を比べて決

定する。また、*rdLVv* (reverse-time *dLVv*) 変換を導入すれば、

- ミウラ変換： $\{b_{2i-1}, b_{2i}\} \mapsto \{x_{2i-1}, x_{2i}\}$
- *rdLVv* 変換： $\{x_{2i-1}, x_{2i}\} \mapsto \{x_{2i-1}^-, x_{2i}^-\}$   

$$\begin{cases} x_{2i-1}^- (1 + \delta^- x_{2i}^-) = x_{2i-1} (1 + \delta x_{2i-2}) \\ x_{2i}^- (1 + \delta^- x_{2i-1}^-) = x_{2i} (1 + \delta x_{2i-1}) \end{cases}$$
- ミウラ逆変換： $\{x_{2i-1}^-, x_{2i}^-\} \mapsto \{b_{2i-1}^-, b_{2i}^-\}$   

$$\begin{cases} b_{2i-1}^- = \frac{1}{\delta^-} (1 + \delta^- x_{2i-2}^-) (1 + \delta^- x_{2i-1}^-) \\ b_{2i}^- = \delta^- x_{2i-1}^- x_{2i}^- \end{cases}$$

によって、下 2 重対角行列  $B^-$  の成分が計算される。ここで、パラメータ  $\delta$ ,  $\delta^-$  は  $1/\delta - 1/\delta^- = \sigma_i^2$  を満たす値とする。なお、 $B^T B$  と  $(B^-)^T B^-$  の成分の符号より  $\eta_k$  を決定する。

以上が *I-SVD* 法の概要である。 $B^+$  の成分以降のツイスト分解の詳細は文献 16) を参照されたい。*dLVv* 変換を用いたツイスト分解を用いることによって、各特異ベクトルは、それぞれ独立に計算可能となる。

ライブラリを開発する際、0 に非常に近い特異値が得られた場合、 $V$  および  $U$  を別々にツイスト分解することによって特異ベクトルを計算すべきである。

### 3. Golub-Kahan-Lanczos 法による上 2 重対角行列作成

数値計算ライブラリの優劣を判定するには、実行時間と精度の評価がそれぞれ必要となる。一般的に、特異ベクトルの精度には、計算された特異ベクトルの直交性が 1 つの指標として扱われる。しかし、特異ベクトルの真値との誤差を評価することも精度の判定上、重要と考える。

特異値分解のテスト行列である 2 重対角行列を作成する方法は 3 種類が知られている。1 つは、ランダムに 2 重対角行列を作成し、多倍長の QRs 法を用いた特異値分解を行うことによって、真値に近い特異ベクトルの分かったテスト行列を生成する方法である。この場合、非常に小規模な問題しか作成できない。2 つ目は、対称 3 重対角行列の固有値問題用のベンチマーク行列を Cholesky 分解する方法である。3 つ目は、Householder 法による 2 重対角化である。多くの場合、Cholesky 分解や Householder 法の誤差が無視できないため、精度を論じるためのテスト行列の生成法として、2 つ目と 3 つ目の方法は不適切である。そこで、新たに Golub-Kahan-Lanczos 法を用いた上 2 重対角テスト行列の作成法が考えられている。この方法では、すべての特異値とその 1 つの特異値に対応する特異ベクトルが与えられたとき、残りの特異ベ

クトルを多倍長で計算し、それらを解として持つ上 2 重対角行列が構成される．この詳細については、別論文にまとめる．

本章では、Golub-Kahan-Lanczos 法によるテスト行列作成法の概略について述べる．

実正方行列  $A$  は、適当な直交行列  $P = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m)$ 、 $Q = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m)$  によって、 $P^T A Q = B$  を満たす

$$B = \begin{pmatrix} b_1 & b_2 & & & \\ & & b_3 & \ddots & \\ & & & \ddots & b_{2m-2} \\ 0 & & & & b_{2m-1} \end{pmatrix} \quad (5)$$

へ変換される．ここで、 $AQ = PB$ 、 $A^T P = QB^T$  に注意すると、 $A^T A Q = A^T P B = QB^T B$  なので、 $A$  と  $B$  の特異値が一致することが分かる．このような  $P$ 、 $Q$  は、Golub-Kahan-Lanczos 法<sup>8)</sup> によって計算される．

特に、 $A$  が対角行列  $\Sigma$  ならば、 $P^T \Sigma Q = B$  より、 $B$  の特異値が  $\Sigma$  の対角成分に並ぶと見なせる．逆に  $\Sigma$  を与えると、 $\Sigma Q = PB$ 、 $\Sigma^T P = QB^T$  を満たす  $P$ 、 $Q$  および  $B$  が決定できる．具体的には、 $\mathbf{q}_1$  を長さ 1 の任意ベクトルとして、

$$\begin{cases} b_1 = \|\Sigma \mathbf{q}_1\|, & \mathbf{p}_1 = \Sigma \mathbf{q}_1 / b_1 \\ \begin{cases} b_{2h} = \|\alpha_h\|, & \mathbf{q}_{h+1} = \alpha_h / b_{2h} \\ b_{2h+1} = \|\beta_h\|, & \mathbf{p}_{h+1} = \beta_h / b_{2h+1} \end{cases} \\ \alpha_h = \Sigma^T \mathbf{p}_h - b_{2h-1} \mathbf{q}_h, & \beta_h = \Sigma \mathbf{q}_{h+1} - b_{2h} \mathbf{p}_h \end{cases} \quad (6)$$

( $h: 1 \leq h \leq m-1$ )

によって、 $P$ 、 $Q$  および  $B$  の成分が逐次的に計算される．つまり、 $\Sigma$  の対角成分を特異値とする  $B$  が作成でき、同時に  $P$  と  $Q$  の行ベクトルが  $B$  の真の特異ベクトルとなる．式 (6) では、近接した 2 数の減算による桁落ちの可能性があるので、テスト行列  $B$  の作成には多倍長演算を使わなければならない．

QRs 法や  $I$ -SVD 法で実装されたライブラリでは、特異値の並びをソートする機能があるため、数値実験の比較の際には、 $\Sigma$  および  $P$ 、 $Q$  の行入れ替えが必要である．

#### 4. 実験結果

密行列の特異値分解では、Golub-Kahan<sup>8)</sup> 以来、計算量の低減のため、まず Householder 法によって上 2 重対角化する方法が主流である<sup>2)</sup>．Householder 法で得られる直交行列と上 2 重対角行列の特異値分解

表 1 計算機性能

Table 1 Performance of each computer.

$C_{P4}$		$C_I$
CPU	Pentium4 2.6 GHz	Itanium2 1.6 GHz
Memory	3 GB	8 GB
L1 D	8 KB	32 KB
L1 I	12 Kμops	32 KB
L2	512 KB	256 KB
OS	Debian 3.0 Linux 2.4.24	RedHut 2.1AS Linux 2.4.24
Compiler	GNU 3.0.4	GNU 3.0.4
$C_O$		$C_{G5}$
CPU	AMD Opteron 2.4 GHz	Power PC G5 2.0 GHz
Memory	2 GB	3.5 GB
L1 D	64 KB	32 KB
L1 I	64 KB	64 KB
L2	1024 KB	512 KB
OS	Fedora Core 3 Linux 2.6.9-1.667	Darwin 8.2.0
Compiler	GNU 3.4.2	GNU 3.4.2

を合成すれば、密行列の特異値分解が完了する．文献 16) において提案された  $I$ -SVD 法も既存の QRs 法も 2 重対角行列に対して特異値分解を行う方法である．そこで、本章では、2 重対角行列をテスト行列として与え、QRs 法と  $I$ -SVD 法を実装したライブラリの計算精度と実行時間について検証する．なお、QRs 法や  $I$ -SVD 法が、上 2 重対角化である Householder 法を含めた密行列の特異値分解においてどのような位置づけになるかを把握するために、密行列の特異値分解の全行程の実行時間と精度についての比較数値実験の結果を付録 A.1 に与える．ただし、付録 A.1 に示された結果は、限られた例に対する実験結果であり、任意の密行列に対する結果ではない．

QRs 法のライブラリとしては、LAPACK の  $DBD$ - $SQR$  を用いた． $I$ -SVD 法のライブラリとしては、逆反復法を付加しない  $DBDSLV.T$  と逆反復法付きの  $DBDSLV.I$  を開発した．さらに、 $DBDSLV.I$  に修正 Gram-Schmidt 法による再直交化を加えた  $DBD$ - $SLV.G$  も作成した． $DBDSLV.*$  において、特異値は、 $mdLVs$  法を実装したライブラリ  $DLVS$ <sup>29),30)</sup> で計算される．この際、 $DLVS$  に関して、シフト量  $S^{(n)}$  の見積りは Johnson 境界を用いた．また、SPLIT や収束による減次を行うかどうかは、 $DBDSQR$  と同様の指標を用いた．特異値計算および特異ベクトル計算において、差分間隔は固定値  $\delta = 1$  とした．数値実験に用いた計算機性能は表 1 のとおりである．GNU コンパイラのオプションには O3 を指定し、各計算機で

表 2 計算機  $C_{P4}$  における誤差  
Table 2 Errors in  $C_{P4}$ .

	Max	Min	Average	deviation
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \bar{\sigma}_i [10^{-12}]$				
<i>DBDSQR</i>	1.10	0.792	0.893	0.0528
<i>DBDSLV_*</i>	0.554	0.219	0.322	0.0727
$\ V - \bar{V}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	56.2	3.84	10.2	6.63
<i>DBDSLV_T</i>	58.0	1.47	6.42	6.86
<i>DBDSLV_I</i>	5.74	0.109	0.423	0.660
$\ U - \bar{U}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	56.1	3.84	10.2	6.63
<i>DBDSLV_T</i>	58.4	1.47	6.62	6.91
<i>DBDSLV_I</i>	5.19	0.113	0.414	0.640
$\ V^T V - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.00867	0.00802	0.00831	0.000128
<i>DBDSLV_T</i>	7.49	0.206	0.673	0.832
<i>DBDSLV_I</i>	0.341	0.00748	0.0324	0.0531
$\ U^T U - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.00859	0.00801	0.00828	0.000118
<i>DBDSLV_T</i>	7.53	0.215	0.750	0.865
<i>DBDSLV_I</i>	0.318	0.00716	0.0315	0.0522
$\ B - UV^T\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.00745	0.00643	0.00690	0.000198
<i>DBDSLV_T</i>	139	2.54	18.9	22.2
<i>DBDSLV_I</i>	4.20	0.0620	0.398	0.583

表 3 計算機  $C_I$  における誤差  
Table 3 Errors in  $C_I$ .

	Max	Min	Average	deviation
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \bar{\sigma}_i [10^{-12}]$				
<i>DBDSQR</i>	1.34	1.04	1.18	0.0729
<i>DBDSLV_*</i>	1.03	0.414	0.656	0.128
$\ V - \bar{V}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	138	5.14	14.0	15.8
<i>DBDSLV_T</i>	107	3.10	11.1	12.5
<i>DBDSLV_I</i>	6.46	0.115	0.430	0.704
$\ U - \bar{U}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	138	5.14	14.0	15.7
<i>DBDSLV_T</i>	109	3.10	11.3	12.6
<i>DBDSLV_I</i>	4.79	0.113	0.397	0.560
$\ V^T V - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0121	0.0114	0.0118	0.000139
<i>DBDSLV_T</i>	16.0	0.357	1.23	1.76
<i>DBDSLV_I</i>	0.542	0.0111	0.0372	0.0644
$\ U^T U - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0120	0.0113	0.0117	0.000134
<i>DBDSLV_T</i>	16.0	0.368	1.31	1.77
<i>DBDSLV_I</i>	0.399	0.0110	0.0346	0.0528
$\ B - UV^T\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0105	0.00903	0.00972	0.000276
<i>DBDSLV_T</i>	1070	4.81	56.2	137
<i>DBDSLV_I</i>	5.95	0.0718	0.390	0.688

異なるバイナリを作成する。 $C_{P4}$  のレジスタは 80 bit であり、その他の計算機のレジスタは 64 bit である。 $C_I$  と  $C_{G5}$  には、和積の融合演算ユニット<sup>1)</sup>が搭載されている。これらの計算機を用いて、レジスタや演算ユニットの違いによる差が生じるかどうかを確認する。

計算精度と実行時間の平均的な結果を得るために、テスト行列として、3 章で述べたテスト行列作成法を利用して、 $[0, 1]$  のランダムな特異値を持つ 1,000 次上 2 重対角行列を 100 個生成する。なお、特異値の集積度の大きな行列や条件数の大きな行列に対する *DBDSLV\_\** の適用例として、付録 A.2 にいくつか実験結果を与える。

以下、4.1 節と 4.2 節では、それぞれ、各ライブラリの計算精度と実行時間に関する実験結果を示し、考察を行う。4.3 節では、*DBDSLV\_I* で計算された特異ベクトルの直交性を向上させる方法について説明し、その効果を数値的に示す。

#### 4.1 計算精度

本節では、*DBDSQR*、*DBDSLV\_T* および *DBDSLV\_I* の計算精度を比較する。そのために、特異値の精度の指標  $\sum_{i=1}^m |\sigma_i - \bar{\sigma}_i| / \bar{\sigma}_i$ 、左右直交行列  $U$  と  $V$  の精度の指標  $\|U - \bar{U}\|_{sum}$ 、 $\|V - \bar{V}\|_{sum}$ 、 $U$  と  $V$  の直交性の度合を表す  $\|U^T U - E\|_{sum}$ 、 $\|V^T V - E\|_{sum}$  および  $\|B - UV^T\|_{sum}$  を調べた。ここで、 $\bar{\sigma}_i$ 、 $\bar{U}$ 、 $\bar{V}$  はそれぞれ真の特異値、左右直交行列とし、 $\|X\|_{sum}$

は  $X$  の要素の絶対値の総和を表すものとする。

100 個のテスト行列を用いた各計算機における結果を表 2、表 3、表 4、表 5 に示す。表中の Average は、100 回の試行における平均値を表す。Max と Min は、それぞれ、最大値と最小値を表す。

80 bit レジスタの影響により、 $C_{P4}$  における精度が他の計算機と比べて良い<sup>29)</sup>。また、*DBDSQR* に関して、和積の融合演算ユニットを利用して計算の一部分を 105 bit で計算することができる<sup>1)</sup>  $C_I$  と  $C_{G5}$  の誤差の総和の平均値は、 $C_O$  に比べて若干良い<sup>29)</sup>。しかし、和積の融合演算ユニットでの計算終了後、各値は 64 bit に変換されるため、テスト行列によっては、誤差の総和が大きくなる<sup>29)</sup>。

*DBDSLV\_\** の特異値は、*DLVS* によって高精度に計算される<sup>29),30)</sup>。ゆえに、すべての計算機において、計算された特異値の相対精度は、*DBDSQR* に比べ、*DBDSLV\_\** のほうが良い。

理論的に、 $V^T V - E$  や  $U^T U - E$  の各要素の値は、0 となる。*DBDSQR* で計算された特異ベクトルにおいて、これらの値は十分に小さく、特異ベクトルの直交性が良い。反面、 $V$  や  $U$  の真値からのずれは大きい。これは、 $B$  の左右から繰り返し適切な直交行列をかけて対角化する *DBDSQR* の計算原理による。つまり、直交行列の積を計算して特異ベクトルを得るため、その直交性は高くなるが、反復計算によって誤

表 4 計算機  $C_O$  における誤差  
Table 4 Errors in  $C_O$ .

	Max	Min	Average	deviation
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \bar{\sigma}_i [10^{-12}]$				
<i>DBDSQR</i>	1.69	1.17	1.32	0.0846
<i>DBDSLV_*</i>	1.03	0.414	0.656	0.128
$\ V - \bar{V}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	119	5.29	15.0	13.8
<i>DBDSLV_T</i>	107	3.11	11.1	12.5
<i>DBDSLV_I</i>	6.44	0.115	0.423	0.699
$\ U - \bar{U}\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	120	5.29	15.0	13.8
<i>DBDSLV_T</i>	108	3.11	11.3	12.6
<i>DBDSLV_I</i>	4.74	0.112	0.395	0.553
$\ V^T V - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0135	0.0127	0.0131	0.000148
<i>DBDSLV_T</i>	16.0	0.358	1.23	1.76
<i>DBDSLV_I</i>	0.532	0.0111	0.0368	0.0632
$\ U^T U - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0134	0.0126	0.0130	0.000139
<i>DBDSLV_T</i>	16.0	0.368	1.31	1.76
<i>DBDSLV_I</i>	0.390	0.0109	0.0348	0.0532
$\ B - U\Sigma V^T\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0115	0.00993	0.0108	0.000294
<i>DBDSLV_T</i>	1070	4.79	56.2	137
<i>DBDSLV_I</i>	5.48	0.0705	0.384	0.647

差が蓄積すると考えられる。

一方、*DBDSLV\_T* および *DBDSLV\_I* において、特異ベクトルの真値に対する誤差は *DBDSQR* より小さい。特に、*DBDSLV\_I* では、*DBDSLV\_T* と比較しても、高精度に特異ベクトルが計算される。これは、逆反復法によって、精度が向上したものである。一般にツイスト分解法では、特異値の近接度によって特異ベクトルの直交性が変化する<sup>10),32)</sup> ため、 $V^T V - E$  や  $U^T U - E$  といった直交性の度合いを示す値の標準偏差（ばらつき）が大きくなると考えられる。

$\|B - U\Sigma V^T\|_{sum}$  の値は、*DBDSLV\_T* と *DBDSLV\_I* のほうが大きく、*DBDSQR* が優位な結果となった。今回の実験では、*DBDSLV\_\** において、 $V$  と  $U$  を独立に計算する方法を採用したが、文献 10) のように同時に  $V$  と  $U$  を計算することも可能で、その場合、 $\|B - U\Sigma V^T\|_{sum}$  はより 0 に近い値となることが期待できる。ただし、文献 10) の方法では、必要最小限の個数の特異ベクトルのみを計算することはできない。

以上より、計算された特異ベクトルの直交性ではテスト行列によって *DBDSQR* に若干劣るが、真値に対する誤差の少ない *DBDSLV\_I* は、必要な特異ベクトルの個数が少ない場合など、応用分野によって非常に有効であると考えられる。

表 5 計算機  $C_{G5}$  における誤差  
Table 5 Errors in  $C_{G5}$ .

	Max	Min	Average	deviation
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \bar{\sigma}_i [10^{-12}]$				
<i>DBDSQR</i>	1.41	1.02	1.17	0.0781
<i>DBDSLV_*</i>	1.19	0.462	0.659	0.135
$V [10^{-8}]$				
<i>DBDSQR</i>	72.7	4.95	13.1	9.49
<i>DBDSLV_T</i>	163	3.45	13.7	22.9
<i>DBDSLV_I</i>	14.4	0.138	0.754	1.67
$U [10^{-8}]$				
<i>DBDSQR</i>	72.6	4.94	13.1	9.49
<i>DBDSLV_T</i>	161	3.40	13.3	18.2
<i>DBDSLV_I</i>	5.79	0.150	0.537	0.871
$\ V^T V - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0121	0.0114	0.0117	0.000151
<i>DBDSLV_T</i>	16.4	0.419	1.36	1.96
<i>DBDSLV_I</i>	1.32	0.0136	0.0643	0.149
$\ U^T U - E\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0120	0.0113	0.0116	0.000146
<i>DBDSLV_T</i>	16.3	0.451	1.59	2.03
<i>DBDSLV_I</i>	0.422	0.0130	0.0434	0.0620
$\ B - U\Sigma V^T\ _{sum} [10^{-8}]$				
<i>DBDSQR</i>	0.0103	0.00896	0.00967	0.000282
<i>DBDSLV_T</i>	753	6.38	48.7	99.5
<i>DBDSLV_I</i>	23.4	0.0708	0.788	3.17

表 6 1,000 次行列の実行時間

Table 6 Computational time for 1,000 dimensional matrices.

	Max(s)	Min(s)	Average(s)	deviation
$C_{P4}$				
<i>DBDSQR</i>	78.51	71.88	74.31	1.28
<i>DBDSLV_T</i>	0.79	0.74	0.76	0.011
<i>DBDSLV_I</i>	1.15	1.06	1.09	0.015
$C_I$				
<i>DBDSQR</i>	117.63	107.66	111.27	1.93
<i>DBDSLV_T</i>	1.13	1.08	1.11	0.010
<i>DBDSLV_I</i>	1.68	1.62	1.65	0.011
$C_O$				
<i>DBDSQR</i>	42.96	39.21	40.55	0.70
<i>DBDSLV_T</i>	0.34	0.32	0.33	0.0040
<i>DBDSLV_I</i>	0.49	0.47	0.48	0.0051
$C_{G5}$				
<i>DBDSQR</i>	103.8	93.8	96.61	1.93
<i>DBDSLV_T</i>	0.63	0.61	0.62	0.0053
<i>DBDSLV_I</i>	0.85	0.83	0.84	0.0045

## 4.2 実行時間

各計算機の実行時間を表 6 に示す。

*DBDSQR* の実行時間は、特異値のみならば *DLVS* の数倍程度<sup>29)</sup> であるにもかかわらず、特異値分解では *DBDSLV\_T* や *DBDSLV\_I* の数百倍にも及ぶ。*DBDSQR* は、 $m$  個以下のスカラー値だけでなく、 $m \times m$  密行列の更新を繰り返すため、計算量が  $O(m^3)$  となる。それに対して、*DBDSLV\_T* と *DBDSLV\_I* では、

表 7 実行時間の推移

Table 7 Transition of computational time.

	<i>DBDSQR</i>	<i>DBDSLV_T</i>	<i>DBDSLV_I</i>
<i>C<sub>P4</sub></i>			
1,000	44.92	0.69	1.13
2,000	432.12	2.83	4.91
3,000	2253.74	6.87	11.27
4,000	10970.31	12.24	19.65
5,000	24249.09	19.32	30.63
6,000	42573.60	27.7	43.73
<i>C<sub>I</sub></i>			
1,000	123.50	1.13	1.66
2,000	1014.68	4.43	6.96
3,000	3703.79	10.05	15.44
4,000	9731.30	17.85	27.55
5,000	14572.94	27.25	41.34
6,000	26260.71	39.77	60.59
<i>C<sub>O</sub></i>			
1,000	41.23	0.33	0.47
2,000	399.30	1.54	2.37
3,000	1261.55	3.27	5.14
4,000	6377.32	6.48	9.71
5,000	5702.10	9.99	15.25
6,000	22915.18	15.01	22.37
<i>C<sub>G5</sub></i>			
1,000	95.71	0.63	0.84
2,000	2025.09	3.23	4.16
3,000	11530.82	7.21	9.43
4,000	23894.04	13.40	17.67
5,000	46545.37	18.68	25.23
6,000	77765.74	28.86	38.07

特異ベクトルを計算するのに、反復計算を必要とせず、 $O(m^2)$  で実行できる。このため、*DBDSLV\_T* と *DBDSLV\_I* は *DBDSQR* よりも高速になると考えられる。

ランダムな 1,000 ~ 6,000 次上 2 重対角行列を生成し、実行した。表 7 は、そのときの実行時間の推移表である。*DBDSLV\_T* および *DBDSLV\_I* において、行列サイズが 2 倍になると実行時間は約 4 倍になり、6 倍になると約 36 倍になることが確認できた。これは、*I-SVD* 法において特異値計算部も特異ベクトル計算部もともに実行時間が  $O(m^2)$  であるという理論的な見積りと一致する。

*DBDSQR* に関して、計算機 *C<sub>I</sub>* を用いた場合、実行時間が  $O(m^3)$  という理論に近い結果を得た。しかし、*C<sub>P4</sub>* では 3,000 次および 4,000 次以降、次数が大きくなる際に、*C<sub>G5</sub>* においては 2,000 次に大きくなる際に、実行時間が理論値以上に増加する結果が得られた。*C<sub>O</sub>* においては、1,000 ~ 3,000 次および 5,000 次の間では、理論通りの実行時間の増加が生じていたのに対して、4,000 次と 6,000 次の次数になる場合は理論値以上に増加した。そのため、*C<sub>O</sub>*

表 8 *DBDSLV\_G* の誤差Table 8 Errors in the case of *DBDSLV\_G*.

	Max	Min	Average	deviation
$\ V - \hat{V}\ _{sum} [10^{-8}]$				
<i>C<sub>P4</sub></i>	8.61	0.103	0.415	0.883
<i>C<sub>I</sub></i>	11.5	0.114	0.454	1.15
<i>C<sub>O</sub></i>	10.8	0.112	0.439	1.09
<i>C<sub>G5</sub></i>	14.0	0.137	0.794	1.75
$\ U - \hat{U}\ _{sum} [10^{-8}]$				
<i>C<sub>P4</sub></i>	8.67	0.101	0.392	0.878
<i>C<sub>I</sub></i>	11.6	0.103	0.432	1.16
<i>C<sub>O</sub></i>	10.9	0.103	0.419	1.09
<i>C<sub>G5</sub></i>	6.84	0.131	0.643	1.22
$\ V^T V - E\ _{sum} [10^{-8}]$				
<i>C<sub>P4</sub></i>	0.000777	0.000735	0.000753	0.00000840
<i>C<sub>I</sub></i>	0.00115	0.00107	0.00112	0.0000156
<i>C<sub>O</sub></i>	0.00116	0.00108	0.00113	0.0000153
<i>C<sub>G5</sub></i>	0.00116	0.00108	0.00111	0.0000168
$\ U^T U - E\ _{sum} [10^{-8}]$				
<i>C<sub>P4</sub></i>	0.000893	0.000832	0.000865	0.0000117
<i>C<sub>I</sub></i>	0.00132	0.00124	0.00129	0.0000159
<i>C<sub>O</sub></i>	0.00134	0.00125	0.00130	0.0000157
<i>C<sub>G5</sub></i>	0.00133	0.00125	0.00129	0.0000157
$\ B - U\Sigma V^T\ _{sum} [10^{-8}]$				
<i>C<sub>P4</sub></i>	0.00155	0.000497	0.000621	0.000137
<i>C<sub>I</sub></i>	0.00204	0.00111	0.00127	0.000128
<i>C<sub>O</sub></i>	0.00204	0.00112	0.00127	0.000126
<i>C<sub>G5</sub></i>	0.00221	0.00112	0.00138	0.000184

に関して、4,000 次の場合のほうが 5,000 次の場合よりも遅くなるという現象がみられた。*DBDSQR* のアルゴリズムの内部では、内積計算が非常に多い。この内積計算にライブラリ BLAS<sup>21)</sup> を用いることによって、キャッシュを有効活用し実行時間を短縮する工夫がなされている。しかし、行列サイズが大きくなると、キャッシュが不足する可能性がある。そのため、*DBDSQR* においては、理論値以上に実行時間が増加する結果が得られたものと考えられる。

以上より、行列サイズに対する良いスケーラビリティを示す *DBDSLV\_T* および *DBDSLV\_I* は大規模行列向きである。また、*DBDSLV\_\** は、必要に応じて数個の特異ベクトルだけをより高速に計算するなどの実用的な対応も可能である。

#### 4.3 直交性の向上

4.1, 4.2 節より、*DBDSLV\_I* は、*DBDSQR* に比べ、非常に高速で、計算された特異値と特異ベクトルに含まれる誤差も少ない。*DBDSLV\_I* で計算された特異ベクトルの直交性は良いこともあるが、多くの場合、*DBDSQR* より劣る。そこで、*DBDSLV\_I* 実行後の再直交化について検討する。ただし、再直交化が可能となるのは、すべての特異ベクトルを計算した場合のみである。



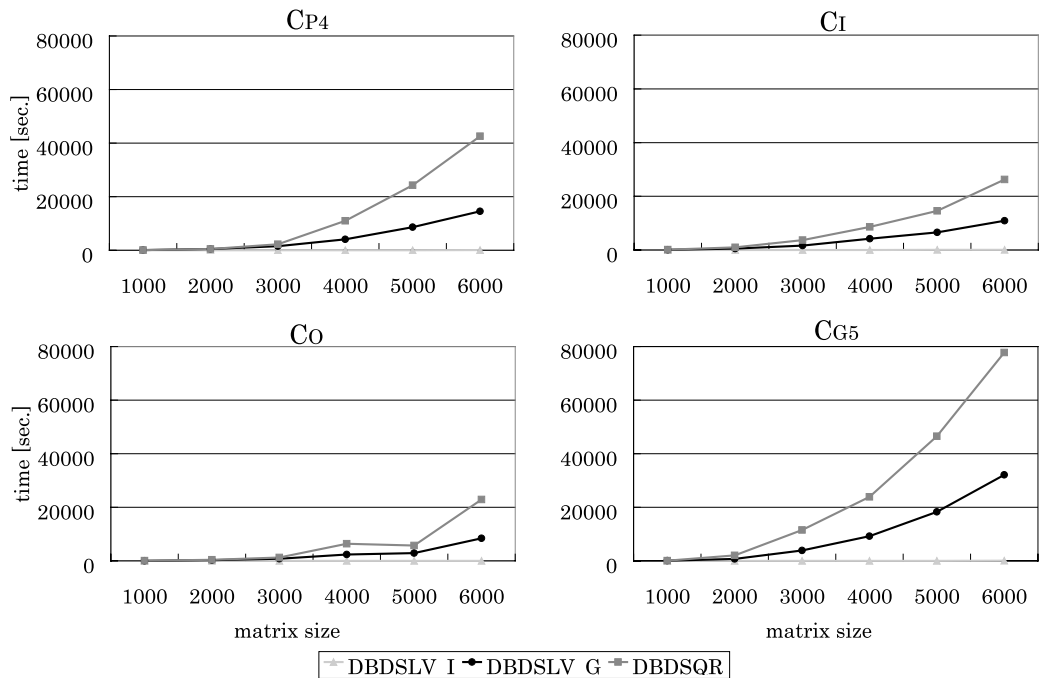


図 2 実行時間の推移グラフ

Fig. 2 Transition graph of computational time.

表 9 *DBDSLV\_G* における 1,000 次行列の実行時間  
Table 9 Computational time for 1,000 dimensional matrices in *DBDSLV\_G*.

	Max(s)	Min(s)	Average(s)	deviation
$C_{P4}$	29.74	25.85	27.28	0.70
$C_I$	39.56	38.30	38.68	0.21
$C_O$	12.60	11.91	12.22	0.14
$C_{G5}$	46.97	46.94	46.96	0.0066

直交化法として、古典的 Gram-Schmidt 法<sup>18)</sup>、修正 Gram-Schmidt 法、Householder 変換を使う方法<sup>33),34)</sup>、Givens 回転を用いる方法などがある。本節では、修正 Gram-Schmidt 法による再直交化計算を *DBDSLV\_I* に加えたライブラリ *DBDSLV\_G* の計算精度と実行時間の評価を行う。

表 8 は、1,000 次行列の特異値分解における計算精度を表す。特異値は、*DBDSLV\_G* においても、*DLVS* によって計算されるため、*DBDSLV\_T* および *DBDSLV\_I* と同精度になる。*DBDSLV\_G* で計算された特異ベクトルの真値からのずれは、*DBDSLV\_I* と同程度であり、*DBDSQR* よりもきわめて小さい。満たすべき直交性は、*DBDSQR* よりも良好である。

表 9 は、1,000 次行列に対する実行時間を表す。*DBDSLV\_G* は再直交化するため *DBDSLV\_I* よりも遅くなるが、*DBDSQR* よりも依然として高速である。

表 10 *DBDSLV\_G* に関する実行時間の推移  
Table 10 Transition of computational time in *DBDSLV\_G*.

	$C_{P4}$	$C_I$	$C_O$	$C_{G5}$
1,000	40.44	54.33	12.04	46.95
2,000	445.62	527.36	293.66	697.04
3,000	1503.01	1650.39	784.67	3888.98
4,000	4090.16	4183.84	2357.22	9218.95
5,000	8613.28	6576.96	2901.11	18264.40
6,000	14523.70	10888.00	8389.14	32132.12

表 10 は、行列サイズの大きさと *DBDSLV\_G* の実行時間の関係を表す。図 2 は、1,000 ~ 6,000 次行列における各ルーチンの実行時間の推移グラフである。再直交化のための実行時間は、 $O(m^3)$  である。そのため、行列サイズが大きくなるにつれ、*DBDSQR* と同様の傾向で実行時間の増加がみられる。また、*DBDSLV\_G* においても、行列サイズが大きくなるとキャッシュが足りなくなり、実行時間が  $O(m^3)$  の割合以上に増加する現象が見られた。しかし、*DBDSQR* に比べ、*DBDSLV\_G* の実行時間の増加傾向は緩やかであるため、より大規模な行列の特異値分解が可能であると考えられる。

以上より、*DBDSLV\_G* は、*DBDSQR* よりも真値に近い特異値と特異ベクトルが高速に計算でき、得られた特異ベクトルの直交性までも *DBDSQR* を上回る。

## 5. ま と め

本論文では、特異値分解法 *I-SVD* のライブラリ *DBDSLV* を開発し、その計算精度と実行時間について論じた。また、*DBDSLV* で計算された特異値と特異ベクトルの真値に対する誤差を正しく見積もるために、Golub-Kahan-Lanczos 法を利用して、真の特異値と特異ベクトルが長桁まで正確に分かる上 2 重対角行列の作成方法を用いた。

代表的な特異値分解ライブラリ *DBDSQR* は、*DBDSLV\_T* や *DBDSLV\_I* よりも特異ベクトルの直交性は良いが、計算量が多く、真値からのずれも大きかった。一方、*DBDSLV\_T* と *DBDSLV\_I* は、特異ベクトルの直交性が *DBDSQR* より若干劣るものの、真値からのずれは小さく、非常に高速であった。また、*DBDSQR* と違って、*DBDSLV\_\** は状況に応じて実行時間とメモリ使用量を効率化できる動作原理を持つ。特に、必ずしもすべての特異ベクトルを計算する必要はない分野において、*DBDSLV\_T* と *DBDSLV\_I* は有用な特異値分解ライブラリといえよう。

特異ベクトルの直交性を重視するならば、*DBDSLV\_I* を実行後、再直交化を実行すればよい。*DBDSLV\_I* で計算された特異ベクトルの直交性は、たとえば修正 Gram-Schmidt 法を利用すれば、大幅に改善され、その高い直交性は *DBDSQR* をしのぐ。すなわち、*DBDSLV\_G* を利用すれば、*DBDSQR* よりも短時間で直交性にも優れた特異値分解が実現できる。

今後の課題として、様々な計算機やコンパイラが *DBDSLV* に与える影響に関して詳しく調べる必要がある。

## 参 考 文 献

- 1) Agarwal, R.C., Enekel, R.F., Gustavson, F.G., Kothari, A. and Zubair, M.: Fast pseudorandom-number generators with modulus  $2^k$  or  $2^{k-1}$  using fused multiply-add, *IBM J. RES. & DEV.*, Vol.46, No.1, pp.97–116 (2002).
- 2) Demmel, J.: *Applied Numerical Linear Algebra*, SIAM, Philadelphia (1997).
- 3) Demmel, J. and Kahan, W.: Accurate singular values of bidiagonal matrices, *SIAM J. Sci. Sta. Comput.*, Vol.67, pp.191–229 (1994).
- 4) Dhillon, I.S.: A New  $O(n^2)$  Algorithm for the Symmetric Tridiagonal Eigen value/Eigenvector Problem, Ph.D. Thesis, Computer Science Division, University of California, Berkeley, California (May 1997).
- 5) Dhillon, I.S. and Parlett, B.N.: Orthogonal eigenvectors and relative gaps, *SIAM J. Matrix Anal. Appl.*, Vol.25, No.3, pp.858–899 (2004).
- 6) Fernando, K.V. and Parlett, B.N.: Accurate singular values and differential qd algorithms, *Numer. Math.*, Vol.67, pp.191–229 (1994).
- 7) Francis, J.G.F.: The QR transformation—a unitary analogue to the LR transformation—part 1, *Computer J.*, Vol.4, pp.265–271 (1961).
- 8) Golub, G. and Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix, *SIAM J. Numer. Anal.*, Vol.2, pp.205–224 (1965).
- 9) Golub, G. and Reinsch, C.: Singular value decomposition and least squares solutions, *Numer. Math.*, Vol.14, pp.403–420 (1970).
- 10) Grosser, B. and Lang, B.: An  $O(n^2)$  algorithm for the bidiagonal SVD, *Lin. Alg. Appl.*, Vol.358, pp.45–70 (2003).
- 11) Henrici, P.: *Applied and Computational Complex Analysis Volume I*, Wiley-Interscience Publishing, New York (1988).
- 12) Iwasaki, M.: Studies of Singular Value Decomposition in Terms of Integrable Systems, Ph.D. Thesis, Kyoto University (2004).
- 13) Iwasaki, M. and Nakamura, Y.: On the convergence of a solution of the discrete Lotka-Volterra system, *Inverse Problems*, Vol.18, pp.1569–1578 (2002).
- 14) Iwasaki, M. and Nakamura, Y.: Accurate computation of singular values in terms of the shifted integrable scheme (2005). (preprint)
- 15) 岩崎雅史, 中村佳正: 特異値計算アルゴリズム dLV の基本性質について, 応用数学会論文誌, No.15, Vol.3, pp.287–306 (2005).
- 16) 岩崎雅史, 阪野真也, 中村佳正: 実対称 3 重対角行列の高精度ツイスト分解とその特異値分解への応用, 応用数学会論文誌, No.15, Vol.3, pp.461–481 (2005).
- 17) Johnson, C.R.: A Gersgorin-type lower bound for the smallest singular value, *Lin. Alg. Appl.*, Vol.112, pp.1–7 (1989).
- 18) 片桐孝洋, 金田康正: 並列固有値ソルバーの実現とその並列性の改良, 情報処理学会並列処理シンポジウム論文集, pp.223–230 (1998).
- 19) 北 研二, 津田和彦, 獅々堀正幹: 情報検索アルゴリズム, 共立出版 (2002).
- 20) Kublanovskaya, V.N.: On some algorithms for the solution of the complete eigenvalue problem, *Zh. Vychisl. Mat.*, Vol.1, pp.555–570 (1961).
- 21) LAPACK. <http://www.netlib.org/lapack>
- 22) 中川 徹, 小柳義夫: UP 応用数学選書 7 最小二乗法による実験データ解析, 東京大学出版 (1999).
- 23) 中村佳正 (編): 可積分系の応用数理, 裳華房

- (2000).
- 24) 中村佳正, 岩崎雅史: シフトつき可積分特異値分解アルゴリズムについて, 日本応用数学会 2004 年度年会講演予稿集, pp.400-401 (2004).
- 25) 大津展之, 栗田多喜夫, 関田 巖: 行動計量学シリーズ 12 パターン認識—理論と応用, 朝倉書店 (1996).
- 26) Parlett, B.N. and Dhillon, I.S.: Fernando's solution to Wilkinsoon's problem: An application of double factorization, *Lin. Alg. Appl.*, Vol.267, pp.247-279 (1997).
- 27) Parlett, B.N. and Marques, O.A.: An Implementation of the dqds Algorithm (Positive Case), *Proc. International Workshop on Accurate Solution of Eigenvalue Problems*, University Park, PA, 1998, *Lin. Alg. Appl.*, Vol.309, No.1-3, pp.217-259 (2000).
- 28) Rutishauser, H.: Der Quotienten-Differenzen-Algorithmus, *Z. Angre. Math. Mech.*, Vol.5, pp.233-251 (1954). (ドイツ語)
- 29) 高田雅美, 岩崎雅史, 木村欣司, 中村佳正: 高精度特異値計算ルーチンの開発とその性能評価, 情報処理学会論文誌: コンピューティングシステム, Vol.46, No.SIG12 (ACS11), pp.299-311 (2005).
- 30) Takata, M., Kimura, K., Iwasaki, M. and Nakamura, Y.: An Evaluation of Singular Value Computation by the Discrete Lotka-Volterra System, *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications*, Vol.II, pp.410-416 (2005).
- 31) 山本哲朗: 数値解析入門 [増訂版], サイエンス社 (2003).
- 32) 山本有作: 密行列固有値解法の最近の発展 (I), 日本応用数学会論文誌, Vol.15, No.2, pp.181-208 (2005).
- 33) 山本有作, 猪貝光祥, 直野 健: 共有メモリ型並列計算機向けの高並列固有ベクトル解法と SR8000 での評価, 情報処理学会論文誌, Vol.42, No.4, pp.771-778 (2001).
- 34) Yamamoto, Y., Igai, M. and Naono, K.: A new BLAS-3 based parallel algorithm for computing the eigenvectors of real symmetric matrices, *High Performance Scientific and Engineering Computing — Hardware/Software Support*, Yang, L.T. and Pan, Y. (Eds), Kluwer Academic Publishes (2004).

## 付 録

### A.1 ある密行列に対する実験結果

4章では, 上 2 重対角行列に対する実験結果を考察した. 本付録では, 我々のライブラリに Householder 変換による前処理と行列の積を計算する後処理を付加

して, いくつかの密行列に適用した例を与える. その例に対する性能を調べるために, 計算機  $C_{P4}$  において実験を行う.

A.1.1 において, 特異値と特異ベクトルの真値が判明している密行列を用いて実験を行い, 精度を調べる. A.1.2 では, 密行列に対する実行時間について考察する.

#### A.1.1 密行列に対する精度

##### $m$ 次 3 重対角行列

$$A_1^{-1} = \begin{pmatrix} \beta & \gamma & & 0 \\ \alpha & \beta & \gamma & \\ & \alpha & \beta & \ddots \\ & & \ddots & \ddots & \gamma \\ 0 & & & \alpha & \beta \end{pmatrix} \quad (7)$$

の固有値  $\lambda_i^{-1}$  と固有ベクトル  $\mathbf{x}_i$  は以下のように与えられる.

$$\lambda_i^{-1} = \beta + 2\text{sgn}(\gamma)\sqrt{\alpha\gamma}\cos\frac{i\pi}{m+1},$$

$$\mathbf{x}_i = \left( \sin\frac{i\pi}{m+1}, \dots, \left(\sqrt{\frac{\alpha}{\gamma}}\right)^{m-1} \sin\frac{mi\pi}{m+1} \right)^\top. \quad (8)$$

この行列  $A_1^{-1}$  の逆行列  $A_1$  は密行列となる. このとき,  $A_1$  の固有値は  $A_1^{-1}$  の固有値の逆数となり, 固有ベクトルは変わらない. 特に,  $\alpha = \gamma = -1$  かつ  $\beta = 2$  の場合, 逆行列  $A_1$  は,

$$A_1 = \frac{1}{m+1} * \begin{pmatrix} m & m-1 & \cdots & 2 & 1 \\ m-1 & 2(m-1) & \cdots & 2*2 & 2 \\ m-2 & 2(m-2) & \cdots & 2*3 & 3 \\ m-3 & 2(m-3) & \cdots & 2*4 & 4 \\ \vdots & & \ddots & & \vdots \\ 2 & 2*2 & \cdots & 2(m-1) & m-1 \\ 1 & 2 & \cdots & m-1 & m \end{pmatrix} \quad (9)$$

のように密な対称行列となり, 固有値は  $\lambda_i = 1/(4\sin^2(i\pi/2(m+1)))$  となる.

式 (9) の行列  $A_1$  の特異値  $\sigma_i$  と特異ベクトル  $\mathbf{u}_i$ ,  $\mathbf{v}_i$  は, それぞれは,  $\sigma_i = \lambda_i$ ,  $\mathbf{u}_i = \mathbf{v}_i = \mathbf{x}_i$  となる.

表 11 は, 対称な密行列  $A_1$  をテスト行列として与えた場合の相対精度の総和を表す. どのライブラリにおいても, 特異値の誤差の総和は小さい. また, どのラ

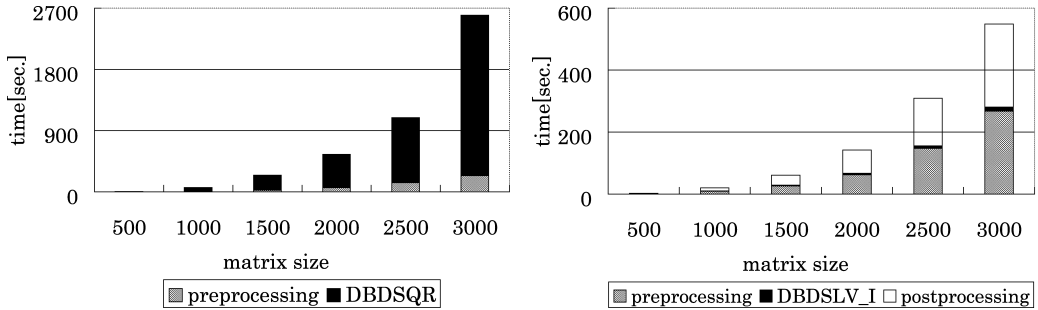


図 3 密行列  $A_2$  に対する実行時間の分布  
Fig. 3 Distribution of computational time in full matrices  $A_2$ .

表 11 密行列  $A_1$  に対する誤差 ( $m = 1,000$ )  
Table 11 Errors in full matrix  $A_1$ . ( $m = 1,000$ )

DBDSQR	DBDSLV-T	DBDSLV-I	DBDSLV-G
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \sigma_i [10^{-10}]$			
1.87780	1.87648	1.87648	1.87648
$\ V - \bar{V}\ _{sum} [10^{-5}]$			
1.09780	1.09748	1.09761	1.09761
$\ U - \bar{U}\ _{sum} [10^{-5}]$			
1.09802	1.09770	1.09783	1.09784
$\ V^T V - E\ _{sum} [10^{-9}]$			
0.09170	1.16373	1.18489	0.06233
$\ U^T U - E\ _{sum} [10^{-9}]$			
0.08082	1.06002	1.54648	0.04383
$\ A_1 - U\Sigma V^T\ _{sum} [10^{-5}]$			
59.5125	9.58206	9.33736	9.42445

イブラリについても、特異ベクトルの要素あたりの最大誤差は  $10^{-10}$  であり、その結果、 $\|U - \bar{U}\|_{sum}$  および  $\|V - \bar{V}\|_{sum}$  は  $10^{-5}$  と大きい。上 2 重対角行列の特異値分解ライブラリを変更しても大きな差がみられないので、大きな誤差の主因は前処理の Householder 法にあると考えられる。

特異ベクトルの直交性は、DBDSQR と DBDSLV\_G が優れている。ところが、DBDSQR において、特異ベクトルの誤差の総和を表す  $\|U - \bar{U}\|_{sum}$  および  $\|V - \bar{V}\|_{sum}$  が大きいいため、 $\|A_1 - U\Sigma V^T\|_{sum}$  の値が大きくなった。さらに、DBDSLV\_G では、 $\|A_1 - U\Sigma V^T\|_{sum}$  の値が、DBDSQR よりも良好な結果となった。これは、特異ベクトルの直交性を重視する DBDSQR とは異なり、特異ベクトルの誤差も小さくなる DBDSLV\_\* の特徴が活かされたものと考えられる。

A.1.2 密行列に対する実行時間

密行列  $A$  の特異値分解には、前処理として Householder 法による上 2 重対角化  $B = U_A^T A V_A$  を実行し、DBDSQR および DBDSLV\_\* を利用する。DBDSQR は、行列  $B$  および  $U_A, V_A$  を入力とし、行列  $A$  の特異値および特異ベクトルを出力とする。一方、DBDSLV\_\* は、行列  $B$  を特異値分解した後、後処理

表 12 密行列  $A_2$  に対する実行時間の一部 [sec.]  
Table 12 A part of computation time in full matrix  $A_2$ . [sec.]

m	500	1000	1500	2000	2500	3000
前処理	0.98	8.89	26.07	61.79	147.00	267.82
後処理	1.10	10.15	31.63	75.32	153.85	268.63

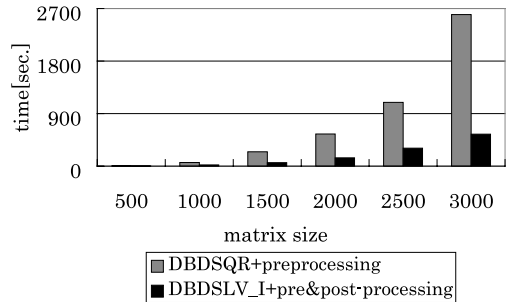


図 4 密行列  $A_2$  の特異値分解の実行時間  
Fig. 4 Computation time for SVD of full matrices  $A_2$ .

として行列積  $U_A U$  および  $V_A V$  の計算が必要である。

実行時間を調べるために、すべての要素をランダムに与えて密行列  $A_2$  を作成した。表 12 は、密行列  $A_2$  の次数を 500 ~ 3,000 次に推移させた場合の前処理と後処理の実行時間を表す。前処理も後処理も  $O(m^3)$  の実行時間を必要とする。図 3 の左図は上 2 重対角行列の特異値分解に DBDSQR を用いた場合、右図は DBDSLV\_I を用いた場合の前処理と後処理をあわせた総実行時間の推移を表す。上 2 重対角行列の特異値分解を DBDSQR で実行すると、総実行時間に占める前処理の割合が小さい。一方、DBDSLV\_I では、2 重対角行列を特異値分解する時間が非常に少なく、前処理と後処理が支配的となる。そのため、図 4 に示すように、行列の次数が大きくなれば、DBDSLV\_I に前処理と後処理を加えても、DBDSQR の実行時間より短くなった。

A.2 悪条件な 2 重対角行列に対する一例

4 章では、ランダムに分布する特異値を持つ比較的

表 13 近接特異値を持つテスト行列

Table 13 Test matrices with adjacent singular values.

	range of singular values
$B_1$	1.0 ~ 2.0
$B_2$	$1.0 \sim 1.0 \times 10^{-2}$
$B_3$	$1.0 \sim 1.0 \times 10^{-4}$
$B_4$	$1.0 \sim 1.0 \times 10^{-6}$

表 14 テスト行列  $B_1 \sim B_4$  に対する精度Table 14 Error in test matrices  $B_1, B_2, B_3$  and  $B_4$ .

	DBDSQR	DBDSLV.T	DBDSLV.I	DBDSLV.G
$\sum_{i=1}^m  \sigma_i - \bar{\sigma}_i  / \bar{\sigma}_i [10^{-13}]$				
$B_1$	6.85	0.911	0.911	0.911
$B_2$	8.64	0.317	0.317	0.317
$B_3$	7.31	0.484	0.484	0.484
$B_4$	7.00	0.517	0.517	0.517
$\ V - \bar{V}\ _{sum}$				
$B_1$	$1.67 \times 10^{-7}$	$1.97 \times 10^{-8}$	$1.46 \times 10^{-9}$	$1.45 \times 10^{-9}$
$B_2$	$1.54 \times 10^{-5}$	$6.91 \times 10^{-7}$	$1.51 \times 10^{-7}$	$1.51 \times 10^{-7}$
$B_3$	$1.68 \times 10^{-3}$	$1.12 \times 10^{-4}$	$1.23 \times 10^{-5}$	$1.23 \times 10^{-5}$
$B_4$	0.208	0.0119	$1.30 \times 10^{-3}$	$1.30 \times 10^{-3}$
$\ U - \bar{U}\ _{sum}$				
$B_1$	$1.67 \times 10^{-7}$	$1.97 \times 10^{-8}$	$1.46 \times 10^{-9}$	$1.45 \times 10^{-9}$
$B_2$	$1.54 \times 10^{-5}$	$6.91 \times 10^{-7}$	$1.51 \times 10^{-7}$	$1.51 \times 10^{-7}$
$B_3$	$1.68 \times 10^{-3}$	$1.12 \times 10^{-4}$	$1.23 \times 10^{-5}$	$1.23 \times 10^{-5}$
$B_4$	0.208	0.0119	$1.31 \times 10^{-3}$	$1.30 \times 10^{-3}$
$\ V^T V - E\ _{sum}$				
$B_1$	$9.83 \times 10^{-11}$	$2.20 \times 10^{-9}$	$1.59 \times 10^{-11}$	$6.67 \times 10^{-12}$
$B_2$	$9.93 \times 10^{-11}$	$7.39 \times 10^{-8}$	$1.69 \times 10^{-11}$	$6.52 \times 10^{-12}$
$B_3$	$9.89 \times 10^{-11}$	$1.11 \times 10^{-5}$	$1.81 \times 10^{-11}$	$6.49 \times 10^{-12}$
$B_4$	$9.54 \times 10^{-11}$	$1.22 \times 10^{-3}$	$3.59 \times 10^{-8}$	$6.77 \times 10^{-12}$
$\ U^T U - E\ _{sum}$				
$B_1$	$9.76 \times 10^{-11}$	$2.19 \times 10^{-9}$	$1.32 \times 10^{-11}$	$5.78 \times 10^{-12}$
$B_2$	$9.64 \times 10^{-11}$	$7.39 \times 10^{-8}$	$1.64 \times 10^{-11}$	$5.39 \times 10^{-12}$
$B_3$	$9.23 \times 10^{-11}$	$1.11 \times 10^{-5}$	$1.38 \times 10^{-11}$	$5.37 \times 10^{-12}$
$B_4$	$9.05 \times 10^{-11}$	$1.22 \times 10^{-3}$	$3.94 \times 10^{-8}$	$5.57 \times 10^{-12}$
$\ B_i - U \Sigma V^T\ _{sum}$				
$B_1$	$2.15 \times 10^{-10}$	$1.60 \times 10^{-7}$	$3.48 \times 10^{-9}$	$1.21 \times 10^{-11}$
$B_2$	$1.46 \times 10^{-10}$	$3.16 \times 10^{-6}$	$1.58 \times 10^{-7}$	$7.83 \times 10^{-12}$
$B_3$	$1.40 \times 10^{-10}$	$5.32 \times 10^{-4}$	$1.32 \times 10^{-5}$	$7.96 \times 10^{-12}$
$B_4$	$1.40 \times 10^{-10}$	0.0732	$1.37 \times 10^{-3}$	$8.06 \times 10^{-12}$

良性と思われるテスト行列を用いて実験を行った。本付録では、計算機  $C_{P4}$  において、悪条件行列のいくつかを特異値分解し、その精度を調べる。

A.2.1 において、近接特異値を持つ行列について精度を調べる。A.2.2 では、条件数の大きな行列の特異値分解について考察する。なお、倍精度演算で近似的に重複特異値を持つといったきわめてシビアな状況下ではより深い議論が別途必要となり、かつ、重根用のサブルーチンも新たに必要とするため、本付録では対象外とした。

#### A.2.1 近接特異値を持つ 2 重対角行列

3 章で述べた Golub-Kahan-Lanczos 法による上 2 重対角行列の作成法を用いて、近接特異値を持つ 1,000

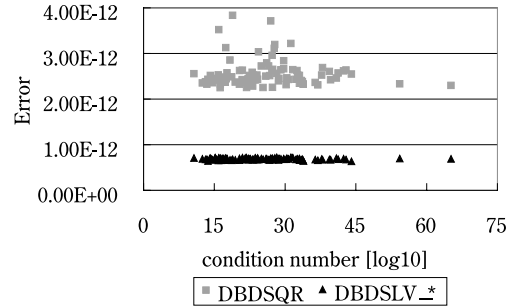


図 5 条件数の大きい行列における特異値の精度

Fig. 5 Error of computed singular values of matrices with large condition number.

次のテスト行列を作成した。表 13 は、各テスト行列の特異値の範囲を表す。表 13 の行列を用いて実験を行った結果を表 14 に示す。どのライブラリにおいても特異値の相対誤差の総和  $\sum_{i=1}^m |\sigma_i - \bar{\sigma}_i| / \bar{\sigma}_i$  は、特異値の近接度に関係なく、ほぼ一定であった。一方、特異値の近接度が上がるほど、特異ベクトルの誤差の総和  $\|U - \bar{U}\|_{sum}$ 、 $\|V - \bar{V}\|_{sum}$  は増加している。特異値および特異ベクトルは、DBDSQR より DBDSLV\_\* のほうが少ない誤差で計算できる。

特異ベクトルの直交性に関して、DBDSLV\_G および DBDSQR を用いた場合、特異値の近接度に関係なく良好であった。しかし、特異ベクトルの真値に対する誤差は、DBDSQR のほうが DBDSLV\_\* より悪い。ゆえに、直交性のみで、特異ベクトルの精度を論じるのは十分ではない。

特異値どうしが近接していても、特異ベクトルの直交性を保つ DBDSQR、特異値・特異ベクトルの真値からの誤差を抑える DBDSLV\_\* という図式は崩れない傾向にある。また、実行時間を考慮しなければ、DBDSLV\_G が最適という結果も 4 章の実験結果と変わらない。

DBDSLV\_T および DBDSLV\_I において、差分間隔  $\delta$  を  $\delta = 1$  に固定すると、特異ベクトルの直交性は、特異値の近接度の上昇に合わせて悪化する。DBDSLV\_\* の差分間隔  $\delta$  を変動すれば、この状況を緩和することは可能である<sup>16)</sup>。ゆえに、悪条件を含む幅広い行列をより正確に特異値分解するには、DBDSLV\_\* の  $\delta$  を可変にすべきである。

#### A.2.2 条件数の大きい行列

3 章で説明した Golub-Kahan-Lanczos 法による上 2 重対角行列の作成法では、条件数の大きな行列を作るのは困難である。そこで、数式処理を使って 1,000 次のテスト行列を作成する。

図 5 は、条件数の大きな上 2 重対角行列の特異値

を  $DBDSQR$  および  $DBDSLVL^*$  で計算したときの相対誤差を表す．横軸は条件数の対数を，縦軸は特異値の相対誤差の総和を表す．近接特異値を持つ行列同様，条件数の大きな行列においても， $DBDSLVL^*$  のほうが  $DBDSQR$  よりも特異値の相対精度が小さかった．また，条件数の大きさと得られた特異値の精度の間に依存関係は確認できなかった．

(平成 17 年 9 月 30 日受付)

(平成 18 年 2 月 21 日採録)



高田 雅美 (正会員)

昭和 52 年生．平成 16 年奈良女子大学大学院人間文化研究科複合領域科学専攻修了．博士 (理学) を同大学より取得．平成 16 年独立行政法人科学技術振興機構戦略的創造研究推進事業の委嘱研究員として，京都大学大学院情報学研究科数理工学専攻数理解析分野にて従事．数値計算ライブラリの開発，分散メモリ環境を対象とする並列プログラムの開発に関する研究に従事．



木村 欣司

昭和 51 年生．平成 16 年神戸大学大学院自然科学研究科博士課程修了．博士 (理学) を同大学より取得．平成 17 年独立行政法人科学技術振興機構戦略的創造研究推進事業の委託研究員として，立教大学理学部数学科にて従事．計算機代数のアルゴリズム開発ならびに実装に従事．日本応用数理学会，日本物理学会，日本数式処理学会各会員．



岩崎 雅史 (正会員)

昭和 49 年生．平成 16 年京都大学大学院情報学研究科博士後期課程修了．博士 (情報学) を同大学より取得．平成 16 年独立行政法人科学技術振興機構戦略的創造研究推進事業の委嘱研究員として，京都大学大学院情報学研究科数理工学専攻数理解析分野にて従事．線形数値計算，微分方程式の漸近解析に関する研究に従事．日本数学会，日本応用数理学会各会員



中村 佳正 (正会員)

昭和 30 年生．昭和 58 年京都大学大学院工学研究科博士課程修了．工学博士を同大学より取得．平成 6 年同志社大学工学部教授．平成 8 年大阪大学大学院基礎工学研究科教授．平成 13 年より京都大学大学院情報学研究科教授．応用可積分系，計算数学の研究に従事．日本応用数理学会，日本数学会，AMS，SIAM 各会員．