

分割統治法とツイスト分解法による新しい特異値分解アルゴリズム

誉田 太朗[†] 高田 雅美^{††}
岩崎 雅史^{†,††} 中村 佳正^{†,††}

本稿では高い並列性を持つ新しい特異値分解アルゴリズムを提案する。近年、ツイスト分解法を採用した固有値・特異値分解アルゴリズムが注目されている。これらは高速ではあるが、特異値計算部の逐次性などが原因で効率的に並列化されていない。そこで我々はこれらと同等の速度・精度に加え、優れた並列性の獲得を目指す。本稿で提案する新しいアルゴリズムは、まず“簡略化した”分割統治法を利用して特異値のみを計算する。この簡略化は、計算時間の多くを費しうる行列演算のほとんどを省略する。次に特異値に対応する特異ベクトルをツイスト分解法により求める。これら各ステップはそれぞれ本質的に並列化可能であるため、このアルゴリズムは高い並列性を持つことが期待される。逐次アルゴリズムを実装し、特異値分布の異なる2種類の行列の特異値分解で評価を行った。

A New Singular Value Decomposition Algorithm by Divide and Conquer and Twisted Factorizations

TARO KONDA,[†] MASAMI TAKATA,^{††} MASASHI IWASAKI^{†,††}
and YOSHIMASA NAKAMURA^{†,††}

This paper proposes a new singular value decomposition algorithm which can be effectively parallelized. For eigen/singular value decomposition, new algorithms with twisted factorization were recently developed. Although these are fast, their parallelism are quite limited due to its seriality in the section of singular value computation. We concern a fully parallelizable algorithm which is in the same level as standard ones with respect to speed and accuracy. Our new algorithm first computes singular values by “compact” D&C, in which singular vectors are not computed. It can be faster than the original one because much of the running time is sometimes consumed for vector updating during singular vector computations. Secondly, the corresponding singular vectors are computed by twisted factorization. The algorithm has great parallelism because each step is executed parallelly. We numerically test it on some SVD computations.

1. はじめに

特異値分解 (SVD) は数値線形代数における最も重要な演算のうちの1つであり、画像処理¹⁾ やデータ検索²⁾ などの多くの分野に応用されている。近年、これらの応用分野におけるデータ量の増大などにより、より高速・高精度なアルゴリズムが求められている。

いくつかの数値計算アルゴリズムは可積分系と関連付けられる。ここでは、線形代数を用いて解くことのできる非線形力学系を可積分系と呼ぶ。いくつかの

保存量を持ち、解は初等関数や特殊関数により書き下せるという可積分系の性質が知られており、これらは適切に離散化された系においても保持される。I-SVD (Integrable system-SVD) は離散可積分系に基づき開発された新しいアルゴリズムであり³⁾⁻⁶⁾、特異値分解を数値安定かつ高速に計算できる。I-SVD はおおきく2つのプロセスで構成される。まず mdLVs (modified discrete Lotka-Volterra) アルゴリズム^{5),7)} により特異値を計算し、次に特異値に対応する特異ベクトルをツイスト分解法^{3),8)-10)} で求める。

並列計算機の利用機会が増えたことや、大きくなるデータ量を処理するため、速度や精度に加えアルゴリズムの並列性への関心が高まっている。特異値計算については、これまでいくつかの並列化手法が提案されている。Parlett らは qd アルゴリズム¹¹⁾ が、 n 個の行列積を $O(\log_2 n)$ で計算する parallel prefix¹²⁾ に

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

^{††} 発展研究, 科学技術振興機構
SORST, JST

^{†††} 奈良女子大学大学院人間文化研究科
Graduate School of Human Culture, Nara Women's University

より並列化できることを示した^{11),13)}。しかしながらこの手法には数値不安定性が報告されている¹⁴⁾だけでなく、大量のプロセッサ間通信が必要とされるため、数値実験での評価はなされていない。一方、我々は領域分割法を適用することで、特異値を計算する dLV (discrete Lotka-Volterra) アルゴリズムの並列化に成功した¹⁵⁾。評価実験の結果、高い並列化効率を持つことが確認された。この領域分割法を用いれば、qd アルゴリズムや QR 法の特異値計算部も同様に並列化可能であると考えられる。ただ 3.1 節で議論するように、アルゴリズムの加速のための“シフト”を導入すると、この領域分割法による効率的な並列化は難しくなってしまう。

一方、QR 法とならび広く利用されている固有値/特異値分解法として分割統治法¹⁶⁾⁻²⁰⁾がある。これは QR 法に比べ高速であることに加え、理論的に高い並列性を持つ。しかしながら行列の固有値/特異値の分布状況によっては計算量が増大する場合がある¹⁸⁾。また空間計算量は $O(n^2)$ であるため大規模問題への適用は難しい。

本稿では以下のように議論を展開し、新しい特異値分解アルゴリズムを提案する。ツイスト分解法は各特異ベクトルを独立に計算するので、I-SVD の並列性は一見優れているように見える。しかし特異値計算部である mdLVs は、dLV を並列化した領域分割法を適用することが難しい。このため実際には I-SVD 全体の並列化効率は限定的となる (3.1 節)。これらの点をふまえ、I-SVD と同等に高速・高精度であり、加えてより優れた並列性を持つ新しい特異値分解アルゴリズムを提案する (3.2 節)。このアルゴリズムはツイスト分解法 (2.1.2 項) および特異値のみを計算する簡略化された分割統治法 (2.2.2 項) を応用する。最後に、速度と精度を評価するため数値実験による比較を行う (4 章)。

2. 特異値分解とアルゴリズム

本章では特異値分解とそのためのアルゴリズムである I-SVD と分割統治法を概説する。

任意の $m \times n$ 行列 A は Housholder 法^{21),22)} を用いて以下のように上 2 重対角行列 B に変換される。

$$A = \hat{U} \begin{pmatrix} B \\ 0 \end{pmatrix} \hat{V}^T \text{ or } \hat{U} \begin{pmatrix} B & 0 \end{pmatrix} \hat{V}^T. \quad (1)$$

ここで \hat{U} と \hat{V} は適当な直交行列である。 B の要素を次のようにおく。

$$B \equiv \begin{pmatrix} b_1 & b_2 & & \\ & b_3 & \ddots & \\ & & \ddots & b_{2n-2} \\ & & & b_{2n-1} \end{pmatrix}, \quad \forall b_j \neq 0. \quad (2)$$

本稿ではこの 2 重対角行列 B の特異値分解を考える。

T_s を以下のように $n \times n$ の 2 重対角行列 B により生成される正定値対称な 3 重対角行列とする。

$$T_s = B^T B. \quad (3)$$

ここで λ_k ($k = 1, \dots, n$) を行列 T_s の固有値とする。下添字の s は対称 (Symmetric) に由来する。そのとき B の特異値 σ_k は

$$\sigma_k = \sqrt{\lambda_k} \quad (4)$$

となる。

2.1 I-SVD アルゴリズム

I-SVD は 2 重対角行列のための特異値分解アルゴリズムであり、可積分系の理論を基に提案された³⁾⁻⁵⁾。I-SVD ではまず mdLVs アルゴリズムにより特異値が最初に計算される。このアルゴリズムは離散化された Lotka-Volterra 系により開発されたものである。次に特異値に対応する特異ベクトルをツイスト分解法で求める。

2.1.1 Lotka-Volterra 系による特異値計算

ここでは特異値計算のための 2 つの手法、dLV とそのシフト版である mdLVs について述べる。

任意の $\delta^{(t)} > 0$ を持つ有限離散 Lotka-Volterra 系は以下のように定義される。

$$\begin{cases} u_k^{(t)} = w_k^{(t-1)} / (1 + \delta^{(t)} u_{k-1}^{(t)}), \\ w_{k-1}^{(t)} = u_{k-1}^{(t)} (1 + \delta^{(t)} u_k^{(t)}). \end{cases} \quad (5)$$

ここで $w_1^{(t)} = w_1^{(t)}$, $w_{2n-1}^{(t)} = u_{2n-1}^{(t)}$, $k = 2, \dots, 2n-1$ である。また $\delta^{(t)}$ は離散化の際の差分間隔とする。初期値を $w_k^{(0)} = b_k$ と選べば、 $t \rightarrow \infty$ において各 $w_{2k-1}^{(t)}$ はそれぞれ特異値 σ_k の 2 乗に収束する²³⁾。これが dLV の基本原理であり、適切な境界条件と初期条件を設定することで特異値を計算できる。

“シフト” は特異値分解アルゴリズムを加速させるための基本的な技法である²²⁾。式 (5) を次のように変形することで dLV にシフトを導入できる。

$$\begin{cases} w_{2k-1}^{(t+1)} = v_{2k-1}^{(t)} + v_{2k-2}^{(t)} - w_{2k-2}^{(t+1)} - S^{(t)}, \\ w_{2k}^{(t+1)} = v_{2k-1}^{(t)} v_{2k}^{(t)} / w_{2k-1}^{(t+1)}. \end{cases} \quad (6)$$

ここで $v_k^{(t)} = u_k^{(t)} (1 + \delta^{(t)} u_{k+1}^{(t)})$ とする。

この dLV のシフト版は mdLVs と呼ばれており^{4),5),7)}, シフト量 $S^{(t)}$ の見積りには Gersgorin 境界²⁴⁾ や Johnson 境界²⁵⁾ が利用されている. 無限精度および有限精度演算の下で mdLVs の収束性と数値安定性が証明されている²⁶⁾.

2.1.2 ツイスト分解法

ここでは, 与えられた特異値に対応する特異ベクトルを計算するツイスト分解法について概説する.

以下の系を考える.

$$(T_s - \hat{\lambda}I)\mathbf{x}^{(k)} = \mathbf{e}_k \gamma_k, \quad \mathbf{x}^{(k)}(k) \equiv 1. \quad (7)$$

ここで \mathbf{e}_k は適切な次元の k 番目の単位ベクトルとし, γ_k は k 番目の方程式の残差ノルムとする. ここで L を以下のような下 2 重対角行列, U を上 2 重対角行列とする.

$$T_s - \hat{\lambda}I = LD^+L^T, \quad (8)$$

$$= UD^-U^T. \quad (9)$$

すなわち L と U は $T_s - \hat{\lambda}I$ の 3 角行列分解により求めることができる. 対角行列 D^+ と D^- の i 番目の要素をそれぞれ D_i^+, D_i^- とし, L と U の i 番目の非対角成分を L_i と U_i する. そのとき $T_s - \hat{\lambda}I$ のツイスト分解は以下のように記述される.

$$T_s - \hat{\lambda}I = N_k D_k N_k^T. \quad (10)$$

この N_k はツイスト行列^{8),9)} と呼ばれる. また D_k は以下のような対角行列である.

$$D_k = \text{diag}(D_1^+, \dots, D_{k-1}^+, \gamma_k, D_{k+1}^-, \dots, D_n^-). \quad (11)$$

もし $\hat{\lambda}$ が固有値 T_s に近い場合には, $\gamma = \arg \min_k |\gamma_k|$ と選ぶことにより $T_s - \hat{\lambda}I$ の特異性が緩和される. $D_r \mathbf{e}_r = \mathbf{e}_r \gamma_r$ および $N_r \mathbf{e}_r = \mathbf{e}_r$ より (10) は次のように変形できる.

$$T_s - \hat{\lambda}I = \gamma_k N_k. \quad (12)$$

これを (7) に代入すると

$$N_k \mathbf{x}^{(k)} = \mathbf{e}_k \quad (13)$$

が得られる. これよりベクトル $\mathbf{x}^{(k)}$ は容易に計算でき, また必要な演算は乗算のみであるため, きわめて高精度であるといえる.

以上よりツイスト分解法が定式化される.

Algorithm: EVVecST

(1) Factor $T_s - \hat{\lambda}I = L_+ D_+ L_+^T$.

(2) Factor $T_s - \hat{\lambda}I = U_- D_- U_-^T$.

(3) Compute and pick an r such that

$$|\gamma_r| = \min_k |\gamma_k|.$$

(4) Solve $N_r^T \mathbf{x} = \mathbf{e}_r$.

(5) Set $\mathbf{x} \leftarrow \mathbf{x}/\|\mathbf{x}\|$.

ツイスト分解の鍵となるのが 3 角行列分解 (8), (9) である. Parlett らは qd 型変種による計算を提案したが^{8),9),27)}, その精度に問題がある場合がある³⁾. この計算に dLV 型変種を用いることで, より高精度な 3 角行列分解を目指す取り組みも進められている³⁾.

2.2 分割統治法とその簡略化

分割統治法はもともと 3 重対角対称行列の固有値問題のために提案された. 基本的には, 問題行列を 2 つの副行列に区分し, 各副行列を固有値分解した結果から元の行列の固有値・固有ベクトルを合成できる, という原理を用いる. 完全 2 分木で問題を構成しながらこの手続きを再帰的に適用するのが分割統治法であり, 本質的に高い並列性を持つ. ここでは Gu と Eisenstat による 2 重対角行列の特異値分解を行う分割統治法を紹介する. あわせて特異ベクトルを計算せず, 特異値のみを求める簡略化された分割統治法について述べる.

2.2.1 2 重対角行列のための分割統治法

$n \times (n+1)$ 上 2 重対角行列 B は $1 < k < n$ である k について以下のように 2 つの行列に分割される.

$$B = \begin{pmatrix} B_1 & 0 \\ b_{2k-1} \mathbf{e}_k^T & b_{2k} \mathbf{e}_1^T \\ 0 & B_2 \end{pmatrix}. \quad (14)$$

ここで B_1 は $(k-1) \times k$ 上 2 重対角行列, B_2 は $(n-k) \times (n-k+1)$ 上 2 重対角行列, b_k は元の行列の要素に対応する適当な値, \mathbf{e}_j は適切な次元を持つ j 番目の単位ベクトルとする. 通常, k は $\lfloor n/2 \rfloor$ とされる. B_i の特異値分解を以下のようにおく.

$$B_i = U_i (D_i \ 0) (V_i \ \mathbf{v}_i)^T. \quad (15)$$

ここで式 (15) を式 (14) に代入すると,

$$B = \begin{pmatrix} 0 & U_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & U_2 \end{pmatrix} \begin{pmatrix} b_{2k-1} \psi_1 & b_{2k-1} \mathbf{1}_1 & b_{2k} \mathbf{f}_2 & b_{2k} \phi_2 \\ 0 & D_1 & 0 & 0 \\ 0 & 0 & D_2 & 0 \end{pmatrix} \times \begin{pmatrix} \mathbf{v}_1 & V_1 & 0 & 0 \\ 0 & 0 & V_2 & \mathbf{v}_2 \end{pmatrix}^T \quad (16)$$

が得られる. ただし $\mathbf{1}_1$ は V_1 の最後の行, ψ_1 は \mathbf{v}_1 の最後の要素, \mathbf{f}_2 は V_2 の最初の行, ϕ_2 は \mathbf{v}_2 の最初の要素とする. Givens 変換を適用し $b_{2k} \phi_2$ を 0 とすると以下ようになる.

$$B = \tilde{U} (M \ 0) (\tilde{V} \ \tilde{\mathbf{v}})^T. \quad (17)$$

ここで

$$\begin{aligned} \tilde{U} &\equiv \begin{pmatrix} 0 & U_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & U_2 \end{pmatrix}, \quad M \equiv \begin{pmatrix} r_0 & b_{2k-1}l_1 & b_{2k}f_2 \\ 0 & D_1 & 0 \\ 0 & 0 & D_2 \end{pmatrix}, \\ \tilde{V} &\equiv \begin{pmatrix} c_0\mathbf{v}_1 & V_1 & 0 \\ s_0\mathbf{v}_2 & 0 & V_2 \end{pmatrix}, \quad \tilde{\mathbf{v}} \equiv \begin{pmatrix} -s_0\mathbf{v}_1 \\ c_0\mathbf{v}_2 \end{pmatrix}, \\ r_0 &= \sqrt{(b_{2k-1}\psi_1)^2 + (b_{2k}\phi_2)^2}, \quad c_0 = \frac{b_{2k-1}\psi_1}{r_0}, \\ s_0 &= \frac{b_{2k}\phi_2}{r_0} \end{aligned} \quad (18)$$

とする。行列 B は \tilde{U} と $(\tilde{V} \quad \tilde{\mathbf{v}})$ により $(M \quad 0)$ へと直交変換される。これらより、 M の特異値分解を行えばよいことになる。この $n \times n$ の行列 M を次のようにおく。

$$M = \begin{pmatrix} z_1 & z_2 & \cdots & z_n \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}. \quad (19)$$

この M の特異値分解は以下の定理により計算できる。

定理 1¹⁹⁾ $U\Sigma V^T$ を M の SVD とする。ただし $U = (\mathbf{u}_1, \dots, \mathbf{u}_n)$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $V = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ とおく。このとき特異値 $\{\sigma_i\}_{i=1}^n$ は

$$0 = d_1 < \sigma_1 < d_2 < \cdots < d_n < \sigma_n < d_n + \|z\|_2 \quad (20)$$

の関係式および secular 方程式

$$f(\sigma) \equiv 1 + \sum_{j=1}^n \frac{z_j^2}{d_j^2 - \sigma^2} = 0 \quad (21)$$

を満たす。そして特異ベクトルは

$$\mathbf{u}_i = \frac{\left(-1, \frac{d_2 z_2}{d_2^2 - \sigma_i^2}, \dots, \frac{d_n z_n}{d_n^2 - \sigma_i^2}\right)^T}{\sqrt{1 + \sum_{k=2}^n \frac{(d_k z_k)^2}{(d_k^2 - \sigma_i^2)^2}}}, \quad (22)$$

$$\mathbf{v}_i = \frac{\left(\frac{z_1}{d_1^2 - \sigma_i^2}, \dots, \frac{z_n}{d_n^2 - \sigma_i^2}\right)^T}{\sqrt{\sum_{k=1}^n \frac{z_k^2}{(d_k^2 - \sigma_i^2)^2}}} \quad (23)$$

により得られる。

行列 M の真の特異値を σ_i 、計算機による近似値を $\hat{\sigma}_i$ とする。定理 1 を使えば理論的には M の真の特異値 σ_i が求められるが、計算機上では誤差を含むため

近似値 $\hat{\sigma}_i$ となる。このとき、たとえ $\hat{\sigma}_i$ が高精度であったとしても比率 $z_k/(d_k^2 - \hat{\sigma}_i^2)$ や $d_k z_k/(d_k^2 - \hat{\sigma}_i^2)$ は、真の比率である $z_k/(d_k^2 - \sigma_i^2)$ や $d_k z_k/(d_k^2 - \sigma_i^2)$ と大きく異なる可能性があり、特異ベクトル計算は数値不安定となりうる。この欠点は以下の定理により克服できる。

定理 2²⁰⁾ 対角行列 $D = \text{diag}(d_1, \dots, d_n)$ と近似特異値 $\{\hat{\sigma}_i\}_{i=1}^n$ は

$$0 \equiv d_1 < \hat{\sigma}_1 < d_2 < \cdots < d_n < \hat{\sigma}_n, \quad (24)$$

を満たすとき、 $\{\hat{\sigma}_i\}_{i=1}^n$ を特異値として持つ行列

$$\hat{M} = \begin{pmatrix} \hat{z}_1 & \hat{z}_2 & \cdots & \hat{z}_n \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix} \quad (25)$$

が存在する。また特異ベクトル $\hat{\mathbf{z}} = (\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n)^T$ は

$$|\hat{z}_i| = \sqrt{(\hat{\sigma}_n^2 - d_i^2) \prod_{k=1}^{i-1} \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_k^2 - d_i^2)} \prod_{k=i}^{n-1} \frac{(\hat{\sigma}_k^2 - d_i^2)}{(d_{k+1}^2 - d_i^2)}}, \quad (26)$$

で与えられる。ただし \hat{z}_i の符号は任意である。

行列 M の近似特異値 $\hat{\sigma}_i$ が計算された後、定理 2 によりその特異値を真の特異値とする行列 \hat{M} を構成する。そして、 \hat{M} の特異ベクトルは $\hat{\sigma}_i$ とベクトル $\hat{\mathbf{z}}$ を式 (22), (23) に代入することで数値安定に求められる。

行列 M の特異値分解が次のとおり求められたとする。

$$M = U_M \Sigma V_M^T. \quad (27)$$

そのとき B の特異値分解は

$$\begin{aligned} B &= \tilde{U} \begin{pmatrix} M & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{\mathbf{v}} \end{pmatrix}^T \\ &= \tilde{U} \begin{pmatrix} U_M \Sigma V_M^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{\mathbf{v}} \end{pmatrix}^T \\ &= U \Sigma V \end{aligned} \quad (28)$$

で得られる。

2.2.2 特異値のための分割統治法

特異ベクトルが不要で特異値のみを計算する場合、分割統治法を非常に簡略化できる²⁰⁾。この場合、副問題の統合のために必要なベクトルのみが計算され、式 (28) にある行列乗算の大部分は省略される。以下の手続きを考え、式 (17), (18) を置き換える。

$$\begin{aligned}
 B &= \tilde{U} \begin{pmatrix} M & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{\mathbf{v}} \end{pmatrix}^T \\
 &= \tilde{U} \begin{pmatrix} U_M \Sigma V_M^T & 0 \end{pmatrix} \begin{pmatrix} \tilde{V} & \tilde{\mathbf{v}} \end{pmatrix}^T \\
 &= U \Sigma \begin{pmatrix} \tilde{V} V_M & \tilde{\mathbf{v}} \end{pmatrix}^T \quad (29) \\
 &= U \Sigma \left(\begin{pmatrix} c_0 \mathbf{v}_1 & V_1 & 0 \\ s_0 \mathbf{v}_2 & 0 & V_2 \end{pmatrix} V_M \begin{pmatrix} -s_0 \mathbf{v}_1 \\ c_0 \mathbf{v}_2 \end{pmatrix} \right)^T
 \end{aligned}$$

これより

$$\begin{aligned}
 \mathbf{f} &= \begin{pmatrix} c_0 \phi_1 & \mathbf{f}_1 & 0 \end{pmatrix} V_M, \\
 \mathbf{l} &= \begin{pmatrix} s_0 \psi_2 & 0 & \mathbf{l}_2 \end{pmatrix} V_M, \\
 \phi &= -s_0 \phi_1, \\
 \psi &= c_0 \psi_2 \quad (30)
 \end{aligned}$$

となる。ここで \mathbf{f}_1 は V_1 の最初の行、 ϕ_1 は \mathbf{v}_1 の最初の要素、 \mathbf{l}_2 は V_2 の最後の行、 ψ_2 は \mathbf{v}_2 の最後の要素、 \mathbf{l} は V の最後の行、 ψ は \mathbf{v} の最後の要素、 \mathbf{f} は V の最初の行、 ϕ は \mathbf{v} の最初の要素とする。

つまり特異値のみを求める分割統治法では、各統合において通常必要となる U_i や V_i の全要素は不要であり、それらの要素の一部である \mathbf{f} 、 \mathbf{l} 、 ϕ 、 ψ と Σ のみを副問題の統合のために計算すればよいといえる。またこれらの要素の計算は (30) の漸化式に帰着されるため、 U_i や V_i の全要素を計算する必要がないことが分かる。本稿ではこの簡素な分割統治法を SVDC (Singular Value oriented Divide and Conquer) と呼ぶ。分割統治法の実行時間の多くが特異ベクトル計算時の行列乗算で費される場合があるため、この部分を省略した SVDC は高速となりうる。加えて $O(n^2)$ が必要であった空間計算量は $O(n)$ とすることができる。

3. 新しい特異値分解アルゴリズム

本章ではまず I-SVD の並列性について論じる。その結果をふまえ、高い並列性を持つ新しい特異値分解アルゴリズムを提案する。

3.1 I-SVD の並列性

はじめに I-SVD の特異値計算部である mdLVs の並列性について考える。この mdLVs のシフトなし版である dLV は、領域分割法により効率的に並列化できる¹⁵⁾。この手法は図 1 に示すように、計算領域を各プロセッサに等分し、その境界において必要な変数を交換することで各領域の独立性を高める。左端のプロセッサを先頭に、各プロセスは時間軸において階段状に進む。ここで領域分割法を同様に mdLVs に適用することを考える。mdLVs では各ステップ t においてシフト量を計算する。この計算には $k = 1, \dots, 2n - 1$

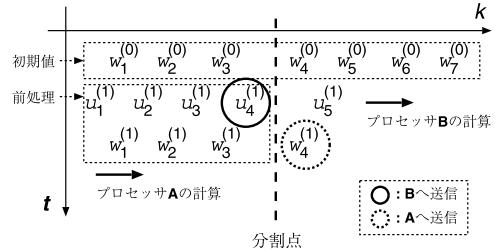


図 1 領域分割法による並列化された dLV の計算 ($n = 4$)
Fig. 1 Parallel dLV process ($n = 4$).

の $w_k^{(t)}$ が必要である。しかし領域分割法により各プロセスは時間軸において階段状になっているため、必要な値のうち一部はその段階では求められていない。加えて、すべての要素を集めてシフト量を逐次計算し、結果をすべてのプロセッサに送信する処理が毎ステップ必要となるため、効率的な並列化は難しい。一方、2.1.2 項にあるとおり、ツイスト分解法はそれぞれのベクトルを独立して計算するため、特異ベクトル計算部は効率良く並列化できるといえる。

以上の議論をふまえ、I-SVD 全体の並列性を考える。 r を並列プログラム中の並列部分の割合とすると、 P 個のプロセッサでの並列化効率 E_P は $E_P = 1/(r + P(1 - r))$ となる (Amdahl の法則²⁸⁾)。これは並列プログラム中の、ある一定の逐次部分がプログラム全体の並列化効率を大きく低下させることを示している。I-SVD のツイスト分解法は各特異ベクトルを独立に計算するが、上述したように mdLVs は並列化が困難である。Amdahl の法則によれば I-SVD 全体の並列化効率は限定的となる可能性がある。実際にツイスト分解法を並列化したものを並列 I-SVD として開発・評価した。計算機には表 1 に記載の Approach の PC クラスタを用いた。表 2 はその実行時間 (上段) と加速率 (下段) の測定例である。行列は対角要素に 2.001、副対角要素に 2.0 を持ち、大きさは 3,000 次とした。ツイスト分解部は理想的に並列化されているが mdLVs 部は逐次実行である。逐次部分の割合 r は約 30% 程度であったが、全体では 16 プロセッサで約 3 倍速という低い効率であった。この行列は近接特異値を持たない良性的行列であり、シフトの効果が薄い悪性的行列であれば mdLVs の実行時間が増加するため、さらに並列化効率は悪化すると予想できる。

3.2 ダブル分割統治法の提案

これまでの議論をふまえ、優れた並列性を持つ高速・高精度な新しい特異値分解アルゴリズムを提案する。

I-SVD は高速なアルゴリズムであり、特にツイスト

表 1 実験に使用した計算機の性能

Table 1 Specification of test computers.

	Appro HyperBlade PC Cluster	Commercial PC (BTO)
CPU	Opteron 1.6 GHz	Opteron 2.4 GHz
Cache	L1D:64 KB	L1D:64 KB
	L1I:64 KB	L1I:64 KB
	L2:1024 KB	L2:1024 KB
Memory	2.0 GB	2.0 GB
Network	Gigabit Ethernet	-
OS	SuSE Linux 8.0 (64 bit)	Fedora Core3 (64 bit)
(Kernel)	(Linux 2.4.19-SMP)	(Linux 2.6.9)
Compiler	pgf77 5.1-3	gcc(g77) 3.4.2
(Option)	-O3	-O3
BLAS	ATLAS 3.6.0	Self complied

表 2 I-SVD アルゴリズムの並列性 (a matrix $n = 3,000$)Table 2 The parallelism of the I-SVD algorithm
(a matrix $n = 3,000$).

Proc.s	1	2	4	8	16
I-SVD	6.43 (1.00)	4.21 (1.53)	3.02 (2.13)	2.47 (2.60)	2.16 (2.98)
mdLVs	1.62 (1.00)	1.62 (1.00)	1.62 (1.00)	1.62 (1.00)	1.62 (1.00)
Twisted	4.46 (1.00)	2.29 (1.95)	1.17 (3.81)	0.59 (7.56)	0.30 (14.87)

Upper: Execution Time [s]

Lower: Parallel Speed-up ratio

分解部は高い計算独立性を持つ。また分割統治法は効率的に並列化でき、また特異値計算に特化し簡略化した SVDC には高速性が期待できる。これら 2 つの長所を採用し、新しいアルゴリズムを構成する。つまり、まず特異値を SVDC で計算し、次に対応する特異ベクトルをツイスト分解法により求める。このアルゴリズムは、(i) 特異値と特異ベクトルを各ステージに分け (divide) 計算して統合する (conquer), (ii) 簡素な分割統治法 (SVDC) で特異値を計算する、という 2 つの意味で分割統治のアプローチを採用していることからダブル分割統治法 (dDC, double Divide and Conquer) と呼ぶ。このアルゴリズムを右に示す。

ダブル分割統治法は以下の性質を持つ。

- (1) 速度．分割統治法が $O(n^2)$ から $O(n^3)$ であるのに対し、SVDC は $O(n \log_2 n)$ の計算量である¹⁸⁾。またツイスト分解法は $O(n^2)$ である²⁷⁾。したがってダブル分割統治法は $O(n^2)$ であり、高速性が期待できる。
- (2) 精度．高精度な分割統治法と同精度で特異値が計算できる。ただし、ツイスト分解法は相対誤差の尺度で高精度に求められた特異値の入力を前提としているが、分割統治法や SVDC に

Algorithm: ダブル分割統治法

- (1) B の特異値 $\hat{\sigma}_k$ を SVDC により計算する。
 - (a) それぞれの大きさが SMLSIZE (通常は 25) 以下となるまで B を再帰的に分割することで、完全二分木として分割統治法の問題ツリーを生成する。
 - (b) 各副問題を特異値分解する。
 $B_i = U_i (D_i \ 0) (V_i \ v_i)^T$ 。
 - (c) 木の各深さにおいて、底の深さから順に各節において、
2 つの副問題を以下のように統合する。
 - secular 方程式により Σ_M と V_M を構成する。
 - f, l, ϕ, ψ を計算する。
(ただし木の根では計算しない)
- (2) 対応する特異ベクトルをツイスト分解法により計算する、
 - (a) For $k = 1, \dots, n$
 $\mathbf{x}_k = \text{EVecST}(B^T B, \hat{\sigma}_k^2)$ を計算する。
 - (b) $V = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ とする。
 - (c) もし $\hat{\sigma}_k \neq 0$ ならば、 $\mathbf{y}_k = B \mathbf{x}_k / \hat{\sigma}_k$ を計算する。
そうでなければ $B^T \mathbf{y}_k = 0$ を解く。
 - (d) $U = (\mathbf{y}_1 / \|\mathbf{y}_1\|, \dots, \mathbf{y}_n / \|\mathbf{y}_n\|)$ とする。

よる特異値は絶対誤差の尺度で高精度であると考えられる。今後、たとえば小さな特異値を持つ行列の計算精度を検証し、必要に応じて補正法などを検討する必要がある。

- (3) 並列性．SVDC およびツイスト分解法はともに効率良く並列化できる。
- (4) 空間計算量．分割統治法は $O(n^2)$ であるのに対し、ダブル分割統治法の SVDC およびツイスト分解法は $O(n)$ である。このため大規模問題にも適用しやすい。
- (5) 一部の特異値/特異ベクトル計算．ツイスト分解法ではそれぞれの特異ベクトルを独立して計算するため、一部の特異ベクトルのみが必要とされている場合には計算時間を節約できる。しかしながら I-SVD の mdLVs はすべての特異値を計算する必要がある。一方、ダブル分割統治法の SVDC は secular 方程式が独立で計算できるため、特異値計算においても計算を省略できる。

これまで SVDC は効果的に利用されていないが、多くの応用分野では特異値だけではなく特異ベクトルも必要とされていることが理由として考えられる。我々はツイスト分解法と組み合わせ、SVDC を特異値分解に活用する。

4. 数値実験による性能評価

本章ではダブル分割統治法を評価するため数値実験を行い、実行時間および計算精度を測定する。いくつかの 2 重対角特異値分解問題を以下の 3 つのルーチンとともに解き、結果を比較する。

- I-SVD : mdLVs の差分間隔は $\delta^{(t)} = 1.0$ とし、ツイスト分解法に加え逆反復法によるベクトル更新を行う。
- D&C : 標準的な分割統治法 (DBDSDC in LAPACK²⁹⁾)。
- QR : シフトつき QR 法 (DBDSQR in LAPACK)。

今回、ダブル分割統治法の SVDC では QR 法で副問題を解く。またツイスト分解は I-SVD のものを用い、最後に逆反復法によりベクトルを更新する。計算機は表 1 に記載の Opteron 2.4 GHz 搭載した PC である。テスト行列は 0 から 1 までのランダムな実数を特異値に持つ 2 重対角行列 (テスト行列 1) と、1 から 2 までのランダムな実数を対角要素と副対角要素に持つ 2 重対角行列 (テスト行列 2) である。これらの乱数は擬似的な一様乱数を用いて生成する。テスト行列 1 に関しては、特異値を生成した後に Golub-Kahan-Lanczos 法^{21),30)} を応用して 2 重対角行列を作成する。このためほぼマシンプシロンの精度で特異値および特異ベクトルの真値が分かっている。この行列を精度の評価に用いる。

実行時間：表 3 にテスト行列 1 の計算時間を示す。10 回試行し平均値を求めた。ダブル分割統治法は I-SVD と同程度で、その他のアルゴリズムよりも非常に高速である。特に分割統治法と比較するとベクトル更新を省略した効果が見える。テスト行列 1 は特異値が一様分布に近いので、分割統治法が高速となるデフレーション²⁰⁾ が起こりにくいと考えられる。行列の

次数に対する計算時間をみると、ダブル分割統治法と I-SVD の計算量はほぼ $O(n^2)$ であることが確認される。表 4 はテスト行列 2 の計算時間である。10 回試行し平均値を求めた。テスト行列 1 に比べ分割統治法が速くなった。これは統合される各副問題の特異値分布が互いに類似していることによるデフレーションが頻繁に発生するためと考えられる。この効果はダブル分割統治法の特異値計算部にも表れ、結果として I-SVD よりも高速となった。このように分割統治法では行列の特異値分布などによって計算時間が大きく異なってしまう。一方、ダブル分割統治法は SVDC で行列演算を省略している分、デフレーションの実行時間に及ぼす影響が比較的小さく、安定的に高速な計算を実現する。

精度：表 5 にテスト行列 1 の計算精度を示す。行列の大きさは $1,000 \times 1000$ とし、10 回試行した平均値を記載した。各値は全要素の相対誤差の和である。ダブル分割統治法と分割統治法は特異値計算 (Σ) では最高精度であった。またベクトルの直交性 ($V^T V - I$, $U^T U - I$) は分割統治法や QR 法に比べやや及ばない。表 6 はテスト行列 2 の計算精度である。テスト行列 1 の場合と同様の傾向を示しているが、特異値分解の精度 ($B - U\Sigma V^T$) は QR 法に比べダブル分割

表 4 要素がランダムな行列の計算時間

Table 4 Execution time of matrix whose elements are random.

	dDC	I-SVD	D&C	QR
$n = 1,000$	0.48	0.56	0.21	40.60
$n = 2,000$	1.94	2.26	0.80	344.45
$n = 3,000$	4.32	5.00	1.74	1131.53

[s]

表 5 特異値がランダムな行列の計算誤差

Table 5 Accuracy of matrix whose singular values are random.

$n = 1,000$	dDC	I-SVD	D&C	QR
Σ	1.6E-13	7.0E-13	1.6E-13	9.0E-13
$V^T V - I$	1.5E-10	1.4E-10	4.0E-11	8.1E-11
$U^T U - I$	1.5E-10	1.4E-10	3.9E-11	8.0E-11
$B - U\Sigma V^T$	2.8E-09	2.2E-09	1.7E-11	7.0E-11

sum of relative errors over all elements

表 6 要素がランダムな行列の計算誤差

Table 6 Accuracy of matrix whose elements are random.

$n = 1,000$	dDC	I-SVD	D&C	QR
$V^T V - I$	1.8e-10	1.8e-10	1.0e-11	7.4e-11
$U^T U - I$	1.0e-09	1.0e-09	9.7e-12	7.4e-11
$B - U\Sigma V^T$	2.1e-10	2.1e-10	1.8e-11	2.6e-10

sum of relative errors over all elements

表 3 特異値がランダムな行列の計算時間

Table 3 Execution time of matrix whose singular values are random.

	dDC	I-SVD	D&C	QR
$n = 1,000$	0.52	0.55	4.18	38.75
$n = 2,000$	2.23	2.13	32.78	348.79
$n = 3,000$	5.08	4.84	107.36	1157.43

[s]

統治法が優れていた。

数値実験より、特異値の分布状況によって大きく実行時間が変動してしまう分割統治法に対して、ダブル分割統治法は安定した高速性を持つことが分かった。また分割統治法が得意とするデフレーションが頻発する行列においては、ダブル分割統治法も高速となることが分かった。精度に関しては、ダブル分割統治法は分割統治法と並び最高精度で特異値を計算した。ベクトルの直交性に関してはツイスト分解を用いているため、分割統治法にはやや及ばなかった。

Glued Wilkinson 行列などの多数の近接固有値やクラスタが存在する行列においてツイスト分解法による直交精度が悪化することが報告されている³¹⁾。今後も数値実験を交えた検証を行う必要がある。また、ツイスト分解法を用いた固有値分解法として MR³ アルゴリズムがある³²⁾。これは dqds アルゴリズム¹³⁾により固有値を計算し、qd 型変種を採用したツイスト分解法により固有ベクトルを計算する。この MR³ を特異値分解に適用したアルゴリズムが開発されており³³⁾、LAPACK4.0 で公開予定である。特異値分布によっては分割統治法よりも高速である場合も報告されており、公開を待って比較実験を行いたい。

5. ま と め

本稿では新しい特異値分解アルゴリズムを提案した。最初に特異値を計算し、次に対応する特異ベクトルをツイスト分解法で求めるといふ、新しいフレームワークが 20 世紀末から注目されはじめている。このツイスト分解法は高い並列性を持つが、特異値計算部はこれまで効率的に並列化が提案されていなかった。しかしアルゴリズム全体の並列性を高めるには、この特異値計算部の並列化が必要不可欠であることを本稿では示した。我々の新しいアルゴリズム(ダブル分割統治法)は、まず簡略化された分割統治法により特異値のみを計算する。そして特異ベクトルをツイスト分解法により求める。このようにすれば分割統治法の計算時間の多くを占める特異ベクトル計算中の行列演算を大幅に省略できるため、行列によっては標準的な分割統治法に比べ高速となる。ダブル分割統治法の各ステップはともに高い並列性が期待できる。加えて $O(n^2)$ である分割統治法の空間計算量に対しダブル分割法は $O(n)$ であるため、大規模問題にも適用できる。数値実験による評価では、特異値分布によって大きく速度が変化する分割統治法に対し、ダブル分割統治法は安定して高速であった。またデフレーションが頻繁に発生する行列では I-SVD よりも高速となった。今後、特

異値のクラスタを持つ悪性の行列などを交えた実験的評価に加え、精度に関する理論的な検証を行う予定である。また MPI³⁴⁾ による並列プログラムを開発し、並列数値計算ライブラリである ScaLAPACK^{35),36)} などとの比較評価を行いたい。あわせてダブル分割統治法を 3 重対角行列の固有値分解に応用したい。

謝辞 本研究にあたり計算機環境の整備に尽力された峯崎征隆氏、ならびにテスト行列の作成にご協力いただいた木村欣司氏に感謝します。

参 考 文 献

- 1) Pratt, W.: *Digital image processing*, Wiley-Interscience Publishing (1978).
- 2) Jiang, F., Kannan, R., Littman, M. and Vempala, S.: Efficient Singular Value Decomposition via Improved Document Sampling, Technical Report CS-99-5, Department of Computer Science, Duke University (1999).
- 3) 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, 日本応用数学会論文誌, Vol.15, No.3 (2005).
- 4) Iwasaki, M.: Studies of Singular Value Decomposition in Terms of Integrable Systems, Ph.D. thesis, Kyoto University (2004).
- 5) Takata, M., Iwasaki, M., Kimura, K. and Nakamura, Y.: An Evaluation of Singular Value Computation by the Discrete Lotka-Volterra System, *Proc. International Conf. on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, USA, Vol.2, pp.410-416 (2005).
- 6) 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).
- 7) Iwasaki, M. and Nakamura, Y.: Accurate computation of singular values in terms of the shifted integrable scheme, preprint (2005).
- 8) Fernando, K.: On computing an eigenvector of a tridiagonal matrix. part 1: basic results, *SIAM J. Matrix. Anal. Appl.*, Vol.18, No.4, pp.1013-1034 (1997).
- 9) Parlett, B. and Dhillon, I.: Fernando's solution to Wilkinson's problem: An application of double factorization, *Lin. Alg. Appl.*, Vol.267, pp.247-279 (1997).
- 10) Dhillon, I. and Parlett, B.: Orthogonal eigenvectors and relative gaps, *SIAM J. Matrix Anal. Appl.*, Vol.25, No.3, pp.858-899 (2004).
- 11) Fernando, K. and Parlett, B.: Accurate singular values and differential qd algorithms, *Nu-*

- mer. Math.*, Vol.67, pp.191–229 (1994).
- 12) Codenotti, B. and Leoncini, M.: *Parallel Complexity of Linear System Solution*, World Scientific (1991).
 - 13) Parlett, B.: The New qd Algorithms, *Acta Numerica*, pp.459–491 (1995).
 - 14) Mathias, R.: The Instability of Parallel Prefix Matrix Multiplication, preprint (1997).
 - 15) 菅田太朗, 高田雅美, 岩崎雅史, 辻本 諭, 中村佳正: ロトカ・ポルテラ系による特異値計算アルゴリズムの並列化, Technical report, 情報処理学会研究報告 HPC-100 (2004).
 - 16) Cuppen, J.: A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numer. Math.*, Vol.36, pp.177–195 (1997).
 - 17) Gu, M. and Eisenstat, S.: A stable and efficient algorithm for the rank-1 modification of the symmetric eigenproblem, *SIAM J. Matrix Anal. Appl.*, Vol.15, No.4, pp.1266–1276 (1994).
 - 18) Gu, M. and Eisenstat, S.: A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, *SIAM J. Matrix Anal. Appl.*, Vol.16, No.1, pp.172–191 (1995).
 - 19) Jessup, E. and Sorensen, D.: A parallel algorithm for computing the singular value decomposition of a matrix, *SIAM J. Matrix Anal. Appl.*, Vol.15, No.2, pp.530–548 (1994).
 - 20) Gu, M. and Eisenstat, S.: A divide-and-conquer algorithm for the bidiagonal SVD, *SIAM J. Matrix Anal. Appl.*, Vol.16, No.1, pp.79–92 (1995).
 - 21) Golub, G. and Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix, *SIAM J. Numer. Anal.*, Vol.2, No.2, pp.205–224 (1965).
 - 22) Demmel, J.: *Applied Numerical Linear Algebra*, SIAM (1997).
 - 23) Iwasaki, M. and Nakamura, Y.: On the convergence of a solution of the discrete Lotka-Volterra system, *Inverse Problems*, Vol.18, No.6, pp.1569–78 (2002).
 - 24) Henrici, P.: *Applied and Computational Complex Analysis*, Vol.1, Wiley-Interscience Publishing (1988).
 - 25) Johnson, C.: A Gersgorin-type lower bound for the smallest singular value, *Lin. Alg. Appl.*, Vol.112, pp.1–7 (1989).
 - 26) 岩崎雅史, 中村佳正: シフトつき可積分スキームによる高精度特異値計算, 日本応用数理学会論文誌 (2005), Submitted.
 - 27) 山本有作: 密行列固有値解法の最近の発展 (I), 日本応用数理学会論文誌, Vol.15, No.2, pp.181–208 (2005).
 - 28) Pacheco, P.: *Parallel Programming with MPI*, Morgan Kaufmann Publishers (1997).
 - 29) Anderson, E., Bischof, Z., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S. and Sorensen, D.: *LAPACK User's Guide, 2nd Edition*, SIAM (1995).
 - 30) 高田雅美, 木村欣司, 岩崎雅史, 中村佳正: 高速特異値分解のためのライブラリ開発, 情報処理学会論文誌コンピューティングシステム (2005), Submitted.
 - 31) Dhillon, I., Parlett, B. and Vömel, C.: Glued matrices and the mrrr algorithm, *SIAM J. Sci. Comput.*, Vol.27, No.2, pp.496–510 (1997).
 - 32) Dhillon, I.: 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 (1997).
 - 33) Willems, P.R., Lang, B. and Vömel, C.: LAPACK WORKING NOTE 166: computing the bidiagonal svd using multiple relatively robust representations, Technical report UCB//CSD-05-1376 Computer Science Division, UC Berkeley, UT-CS-05-551 (2005).
 - 34) Gropp, W., Lusk, E. and Skjellum, A.: *Using MPI*, 2nd Edition, The MIT Press (1999).
 - 35) Blackford, L., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D. and Whaley, R.: *ScaLAPACK Users' Guide*, SIAM (1997).
 - 36) Tisseur, F. and Dongarra, J.: A parallel divide and conquer algorithm for the symmetric eigenvalue problem on distributed memory architectures, *SIAM J. Sci. Comput.*, Vol.20, No.6, pp.2223–2236 (1999).

(平成 17 年 10 月 3 日受付)

(平成 18 年 1 月 29 日採録)



菅田 太朗 (学生会員)

昭和 56 年生・平成 18 年京都大学大学院情報学研究所数理工学専攻修了・同大学院博士後期課程に在籍・平成 16 年より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リサーチスタッフおよび研究補助者として数値計算アルゴリズム, 特に並列化に関する研究に従事・また自身が設立したベンチャー企業にて研究成果の事業化を推進.



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．



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．



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．