

**離散ハングリー戸田方程式に基づく  
Totally Nonnegative 行列に対する固有値計算  
Eigenvalue computation for a totally nonnegative matrix  
based on the discrete hungry Toda equation**

<sup>1)</sup> 福田 亜希子, <sup>2)</sup> 山本 有作, <sup>3)</sup> 岩崎 雅史, <sup>1)</sup> 石渡 恵美子, <sup>4)</sup> 中村 佳正  
<sup>1)</sup> 東京理科大学理学部, <sup>2)</sup> 神戸大学大学院工学研究科, <sup>3)</sup> 京都府立大学生命環境学部,  
<sup>4)</sup> 京都大学大学院情報学研究科

<sup>1)</sup> Akiko Fukuda, <sup>2)</sup> Yusaku Yamamoto, <sup>3)</sup> Masashi Iwasaki,  
<sup>1)</sup> Emiko Ishiwata, <sup>4)</sup> Yoshimasa Nakamura

<sup>1)</sup> Faculty of Science, Tokyo University of Science,

<sup>2)</sup> Graduate School of System Informatics, Kobe University,

<sup>3)</sup> Faculty of Life and Environmental Sciences, Kyoto Prefectural University,

<sup>4)</sup> Graduate School of Informatics, Kyoto University

## 1 はじめに

連続時間な戸田方程式と QR アルゴリズム, 離散戸田方程式と qd アルゴリズムなど, 可積分な方程式と固有値計算アルゴリズムの結び付きは興味深い. 可積分系の 1 つである離散ロトカ・ボルテラ (dLV: discrete Lotka-Volterra) 系からは, 上 2 重対角行列の特異値を求めるための dLV アルゴリズムが提案されている [9]. dLV アルゴリズムに原点シフトを導入した mdLVs (modified dLV with shift) アルゴリズムは, 実行時間, 計算精度ともに実用レベルに達したアルゴリズムといえる [10]. さらに, dLV 系の自然な拡張である離散ハングリーロトカ・ボルテラ (dhLV: discrete hungry Lotka-Volterra) 系

$$\begin{cases} u_j^{(n+1)} = u_j^{(n)} \prod_{p=1}^M \frac{1 + \delta^{(n)} u_{j+p}^{(n)}}{1 + \delta^{(n+1)} u_{j-p}^{(n+1)}}, & j = 1, 2, \dots, M_m, \\ u_{-M}^{(n)} \equiv 0, u_{-M+1}^{(n)} \equiv 0, \dots, u_0^{(n)} \equiv 0, u_{M_m}^{(n)} \equiv 0, u_{M_m+1}^{(n)} \equiv 0, \dots, u_{M_m+M}^{(n)} \equiv 0 \end{cases} \quad (1)$$

からは, 非対称な帯行列の複素固有値を求めるための dhLV アルゴリズムが定式化されている [2]. 非対称な 3 重対角行列に関する数値例によると, dhLV アルゴリズムは LAPACK ルーチンよりも高い相対精度で固有値を求めることができる [4]. また, Totally Nonnegative (TN) 行列と呼ばれる非対称行列が対象のアルゴリズムとして, qd アルゴリズムを拡張した multiple dqd (differential qd) アルゴリズムが提案されている [13]. すべての小行列式が非負である TN 行列は, 組み合わせ論や逆問題など様々な分野に現れる. dhLV アルゴリズムはまた, TN 行列の固有値も求められることが示されている [5].

qd 法の漸化式と等価な離散戸田方程式に対してパラメータ  $M$  を導入すると, 離散ハングリー戸田 (dhToda: discrete hungry Toda) 方程式

$$\begin{cases} Q_j^{(n+M)} = Q_j^{(n)} + E_j^{(n)} - E_{j-1}^{(n+1)}, & j = 1, 2, \dots, m, \\ E_j^{(n+1)} = \frac{Q_{j+1}^{(n)} E_j^{(n)}}{Q_j^{(n+M)}}, & j = 1, 2, \dots, m-1, \\ E_0^{(n)} \equiv 0, E_m^{(n)} \equiv 0 \end{cases} \quad (2)$$

が得られている [12]. 離散ハングリー戸田方程式 (2) から dhToda アルゴリズムと名付けられた TN 行列の固有値を求めるためのアルゴリズムが定式化されている [5]. ここで, multiple dqd アルゴリズム, dhLV アルゴリズム, dhToda アルゴリズムの対象となる TN 行列のクラスを整理しておく. multiple dqd アルゴリズムが対象とする行列は下 2 重対角行列  $L_i$  と上 2 重対角行列  $R_i$  を用いて  $A_{m_L, m_R} = L_1 L_2 \cdots L_{m_L} R_1 R_2 \cdots R_{m_R}$  と表される TN 行列であるが, dhLV アルゴリズム, dhToda アルゴリズムが対象とする行列はそれぞれ TN 行列  $A_{1, M}, A_{M, 1}$  である [5]. 適切な正則行列  $P$  に対して  $A_{M, 1} = P^{-1}(A_{1, M})^T P$  が成り立つことに着目すると, dhLV アルゴリズムと dhToda アルゴリズムの関係, さらに離散ハングリーロトカ・ボルテラ系 (1) と離散ハングリー戸田方程式 (2) を結ぶ Bäcklund 変換も明らかになる [3].

本論文では, TN 行列の固有値を求めるための dhToda アルゴリズムに関する最近の結果を概説する. 2 節では, 減算を含まない differential 型の dhToda アルゴリズムの要点を述べる. 3 節では, 収束加速のために原点シフトを導入した differential 型のシフト付き dhToda アルゴリズムを導く. 4 節では, differential 型のシフト付き dhToda アルゴリズムに対する誤差解析の結果を示す. 5 節では, 原点シフト導入による収束加速を確認し, 誤差解析の結果を裏付けるために数値例を与える. 最後に 6 節でまとめを述べる.

## 2 離散ハングリー戸田方程式に基づく固有値計算アルゴリズム

文献 [5] において, 離散ハングリー戸田方程式 (2) の基本性質のいくつかを明らかにし, これらを利用して TN 行列の固有値を求めるためのアルゴリズムを定式化した. アルゴリズムは離散ハングリー戸田方程式 (2) に基づくため dhToda アルゴリズムと名付けられた. 本節では, dhToda アルゴリズムについて概説する.

qd アルゴリズムの漸化式は, 補助変数を導入して減算のない differential 型と呼ばれる漸化式に変形できる. 初期値が同じであれば, 理論的にはどちらの漸化式を利用しても得られる数列は同じである. ところが, 浮動小数点数演算では, 減算による桁落ちが起こらない分, differential 型の方が有効といえる. qd アルゴリズムの漸化式は離散戸田方程式に他ならず, 離散戸田方程式の拡張版である離散ハングリー戸田方程式 (2) も, 離散戸田方程式に倣って differential 型に書き換えられる. 実際, 新しい変数

$$D_1^{(n)} := Q_1^{(n)}, \quad D_j^{(n)} := Q_j^{(n)} - E_{j-1}^{(n+1)}, \quad j = 2, 3, \dots, m \quad (3)$$

を導入すると, 離散ハングリー戸田方程式 (2) は以下のように変形できる.

$$\begin{cases} Q_j^{(n+M)} = E_j^{(n)} + D_j^{(n)}, & j = 1, 2, \dots, m, \\ E_j^{(n+1)} = \frac{Q_{j+1}^{(n)}}{Q_j^{(n+M)}} E_j^{(n)}, & j = 1, 2, \dots, m-1, \\ D_1^{(n)} = Q_1^{(n)}, \quad D_{j+1}^{(n)} = \frac{Q_{j+1}^{(n)}}{Q_j^{(n+M)}} D_j^{(n)}, & j = 2, 3, \dots, m-1. \end{cases} \quad (4)$$

$Q_j^{(0)} > 0, E_j^{(0)} > 0$  ならば  $Q_j^{(n)} > 0, E_j^{(n)} > 0, n = 1, 2, \dots$  となることは (4) より明らかである. この正

値性を踏まえて  $n \rightarrow \infty$  における  $Q_j^{(n)}, E_j^{(n)}$  に関する漸近挙動を調べると,

$$\lim_{n \rightarrow \infty} \prod_{p=0}^{M-1} Q_j^{(n-p)} = \lambda_j, \quad j = 1, 2, \dots, m, \quad (5)$$

$$\lim_{n \rightarrow \infty} E_j^{(n)} = 0, \quad j = 1, 2, \dots, m-1 \quad (6)$$

が得られる. ただし,  $\lambda_1, \lambda_2, \dots, \lambda_m$  は  $\lambda_1 > \lambda_2 > \dots > \lambda_m > 0$  を満たす実数である.  $Q_1^{(n)}, Q_2^{(n)}, \dots, Q_m^{(n)}$  と  $E_1^{(n)}, E_2^{(n)}, \dots, E_{m-1}^{(n)}$  を成分に含む下 2 重対角行列  $L^{(n)}$  と上 2 重対角行列  $R^{(n)}$  を

$$L^{(n)} := \begin{pmatrix} Q_1^{(n)} & & & & \\ & 1 & & & \\ & & Q_2^{(n)} & & \\ & & & \ddots & \\ & & & & 1 & Q_m^{(n)} \end{pmatrix}, \quad R^{(n)} := \begin{pmatrix} 1 & E_1^{(n)} & & & \\ & 1 & \ddots & & \\ & & & \ddots & \\ & & & & E_{m-1}^{(n)} \\ & & & & & 1 \end{pmatrix} \quad (7)$$

と定めると, 離散ハングリー戸田方程式 (2) の行列表示として,

$$L^{(n+k+M)} R^{(n+k+1)} = R^{(n+k)} L^{(n+k)}, \quad k = 0, 1, \dots, M-1 \quad (8)$$

が得られる. (8) において行列間の等式を成分ごとに書き下すと離散ハングリー戸田方程式 (2) となる. ここで,  $A^{(n)} := L^{(n)} L^{(n+1)} \dots L^{(n+M-1)} R^{(n)}$  とすると,  $A^{(n)}$  は下帯幅が  $M$ , 上帯幅が 1 の下 Hessenberg 行列である.  $Q_j^{(n)} > 0, E_j^{(n)} > 0$  ならば下 2 重対角行列  $L^{(n)}$  と上 2 重対角行列  $R^{(n)}$  は明らかに TN 行列であり, TN 行列の積となる  $A^{(n)}$  もまた TN 行列である. (8) によって  $R^{(n)} A^{(n)} (R^{(n)})^{-1}$  を繰り返し変形すると,

$$\begin{aligned} R^{(n)} A^{(n)} (R^{(n)})^{-1} &= R^{(n)} L^{(n)} L^{(n+1)} \dots L^{(n+M-1)} \\ &= (L^{(n+M)} R^{(n+1)}) \dots L^{(n+M-2)} L^{(n+M-1)} \\ &\quad \vdots \\ &= L^{(n+M)} L^{(n+M+1)} \dots (R^{(n+M-1)} L^{(n+M-1)}) \\ &= L^{(n+M)} L^{(n+M+1)} \dots (L^{(n+2M-1)} R^{(n+M)}) \\ &= A^{(n+M)} \end{aligned} \quad (9)$$

となる. (9) は離散ハングリー戸田方程式 (2) の時間発展  $n \rightarrow n+M$  によって  $A^{(n)}$  から  $A^{(n+M)}$  への相似変形が実行できることを意味する. これが dhToda アルゴリズムの 1 ステップである. 漸近挙動 (5), (6) より,  $\lim_{n \rightarrow \infty} A^{(n)}$  の副対角成分は 0 に収束し,  $\lim_{n \rightarrow \infty} A^{(n)}$  は下三角行列になることが分かる. 下三角行列  $\lim_{n \rightarrow \infty} A^{(n)}$  の対角成分には  $\lambda_1, \lambda_2, \dots, \lambda_m$  が並ぶので,  $A^{(0)}$  と  $\lim_{n \rightarrow \infty} A^{(n)}$  が相似であることを踏まえると,  $\lambda_1, \lambda_2, \dots, \lambda_m$  は  $A^{(0)}$  の固有値であると結論付けられる. つまり, 離散ハングリー戸田方程式 (2) を繰り返し利用すれば,  $A^{(0)}$  の固有値を求めることができる. 以上が, dhToda アルゴリズムの動作原理である. 詳しくは [5] を参照されたい.

### 3 原点シフトの導入

固有値計算アルゴリズムの収束速度を向上させる手段の1つに原点シフトの導入がある. 対称3重対角行列の固有値を求めるためのdqdsアルゴリズムは, differential型のqd (dqd) アルゴリズムに原点シフトを組み込んだアルゴリズムとして知られている. 原点シフト付きのdhTodaアルゴリズムについては[6]で提案したが, dqdsアルゴリズムのように原点シフト導入が容易でないことに注意されたい. 本節では, どのようにdhTodaアルゴリズムにシフトが導入されたかを述べる.

(9)のように $A^{(n)}$ から $A^{(n+M)}$ を求めるdhTodaアルゴリズムの1ステップはLR変換

$$\begin{aligned} A^{(n)} &= \mathcal{L}^{(n)} R^{(n)}, \quad R^{(n)} \mathcal{L}^{(n)} = A^{(n+M)}, \\ \mathcal{L}^{(n)} &:= L^{(n)} L^{(n+1)} \dots L^{(n+M-1)} \end{aligned} \quad (10)$$

と解釈できる. LR変換(10)に対して, 以下のように原点シフトを陰的に導入できる.

$$A^{(n)} - s^{(n)} I = \bar{\mathcal{L}}^{(n)} R^{(n,0)}, \quad A^{(n+M)} = R^{(n,0)} \bar{\mathcal{L}}^{(n)} + s^{(n)} I. \quad (11)$$

ただし,  $s^{(n)}$ をシフト,  $I$ は単位行列とする. ここでは,  $A^{(n)} - s^{(n)} I$ が帯幅 $M$ の下3角行列 $\bar{\mathcal{L}}^{(n)}$ と対角成分がすべて1である上2重対角行列 $R^{(n,0)}$ に破綻なく分解されると仮定する. 対角成分がすべて1である上2重対角行列 $R^{(n,1)}, R^{(n,2)}, \dots, R^{(n,M)}$ を新たに導入し,

$$L^{(n+M+k)} R^{(n,k+1)} = R^{(n,k)} L^{(n+k)}, \quad k = 0, 1, \dots, M-1, \quad (12)$$

$$R^{(n,M)} R^{(n)} = R^{(n+M)} R^{(n,0)} \quad (13)$$

を満たすとする. (12)はLR変換であるのに対して, (13)はRR変換と呼ぶことにする. LR変換(12)とRR変換(13)を施すと,  $R^{(n,0)} A^{(n)} (R^{(n,0)})^{-1}$ は

$$\begin{aligned} R^{(n,0)} A^{(n)} (R^{(n,0)})^{-1} &= (R^{(n,0)} L^{(n)}) L^{(n+1)} \dots L^{(n+M-1)} R^{(n)} (R^{(n,0)})^{-1} \\ &= L^{(n+M)} (R^{(n,1)} L^{(n+1)}) \dots L^{(n+M-1)} R^{(n)} (R^{(n,0)})^{-1} \\ &\quad \vdots \\ &= L^{(n+M)} L^{(n+M+1)} \dots L^{(n+2M-1)} (R^{(n,M)} R^{(n)} (R^{(n,0)})^{-1}) \\ &= A^{(n+M)} \end{aligned} \quad (14)$$

と変形される. (14)のようにLR変換(12)とRR変換(13)の組み合わせで $A^{(n)}$ から $A^{(n+M)}$ を求める操作がシフト付きdhTodaアルゴリズムの1ステップとなる. 上2重対角行列 $R^{(n,k)}$ の上副対角成分を $E^{(n,k)}$ とにおいて(12)の成分間に着目すると,

$$Q_j^{(n+M+k)} = Q_j^{(n+k)} + E_j^{(n,k)} - E_{j-1}^{(n,k+1)}, \quad k = 0, 1, \dots, M-1, \quad (15)$$

$$E_j^{(n,k+1)} = \frac{Q_{j+1}^{(n+k)}}{Q_j^{(n+M+k)}} E_j^{(n,k)}, \quad k = 0, 1, \dots, M-1 \quad (16)$$

が得られる. 同様に, (13)において成分間の等式を書き下すと,

$$E_j^{(n+M)} = E_j^{(n)} + E_j^{(n,M)} - E_j^{(n,0)}, \quad j = 1, 2, \dots, m-1, \quad (17)$$

$$E_{j+1}^{(n,0)} = \frac{E_{j+1}^{(n)}}{E_j^{(n+M)}} E_j^{(n,M)}, \quad j = 1, 2, \dots, m-2 \quad (18)$$

となる. (14) に示した  $A^{(n)}$  から  $A^{(n+M)}$  への相似変形は,  $A^{(n)}$  の成分以外に  $R^{(n,0)}$  の (1, 2) 成分  $E_1^{(n,0)}$  のみ予め求めれば実行できる. 上 2 重対角行列  $R^{(n,0)}$  の上副対角成分すべてが既知でなくてもよいことに注意したい. (11) において (1, 1) 成分と (1, 2) 成分を注視すると,  $E_1^{(n,0)}$  は  $A^{(n)}$  の成分およびシフト量  $s^{(n)}$  によって,

$$E_1^{(n,0)} = \frac{Q_1^{(n)} Q_1^{(n+1)} \cdots Q_1^{(n+M-1)}}{Q_1^{(n)} Q_1^{(n+1)} \cdots Q_1^{(n+M-1)} - s^{(n)}} E_1^{(n)} \quad (19)$$

と与えられることが分かる. (19) によって  $E_1^{(n,0)}$  が得られた後は, (15), (16) によって  $Q_1^{(n+M)}, E_1^{(n,1)}, Q_1^{(n+M+1)}, E_1^{(n,2)}, Q_1^{(n+M+2)}, E_1^{(n,3)}, \dots, Q_1^{(n+2M-1)}, E_1^{(n,M)}$ , 続いて (17), (18) によって  $E_1^{(n+M)}, E_2^{(n,0)}$  の順に求める. 以降, 同様にして (15), (16) によって  $Q_2^{(n+M)}, E_2^{(n,1)}, Q_2^{(n+M+1)}, E_2^{(n,2)}, Q_2^{(n+M+2)}, E_2^{(n,3)}, \dots, Q_2^{(n+2M-1)}, E_2^{(n,M)}$  を, (17), (18) によって  $E_2^{(n+M)}, E_3^{(n,0)}$  を, 最終的には  $E_{m-1}^{(n+M)}, Q_m^{(n+M)}, Q_m^{(n+M+1)}, Q_m^{(n+M+2)}, \dots, Q_m^{(n+2M-2)}, Q_m^{(n+2M-1)}$  まで求めれば,  $L^{(n+M)}, L^{(n+M+1)}, \dots, L^{(n+2M-1)}$ ,  $R^{(n+M)}$  の成分すべてが与えられる. 言い換えれば, (15)–(19) によって  $A^{(n)}$  から  $A^{(n+M)}$  が得られる.

2 節で説明したシフトなしの場合と同じように, シフト付きの場合も differential 型の漸化式を導くことができる [7]. 補助変数  $D_j^{(n)}$  を

$$\begin{aligned} D_1^{(n+k)} &:= Q_1^{(n+k)}, \quad k = 0, 1, \dots, M-1, \\ D_j^{(n+k)} &:= Q_j^{(n+k)} - E_{j-1}^{(n,k+1)}, \quad j = 2, 3, \dots, m, \quad k = 0, 1, \dots, M-1 \end{aligned} \quad (20)$$

と定めると, (15), (16) より  $D_j^{(n+k)}$  から  $D_{j+1}^{(n+k)}$  を求めるための漸化式

$$D_{j+1}^{(n+k)} = \frac{Q_{j+1}^{(n+k)}}{Q_j^{(n+M+k)}} D_j^{(n+k)}, \quad j = 1, 2, \dots, m-1, \quad k = 0, 1, \dots, M-1 \quad (21)$$

が得られる. (15) と (21) を組み合わせると  $Q_j^{(n+M+k)}$  は漸化式

$$Q_j^{(n+M+k)} = D_j^{(n+k)} + E_j^{(n,k)}, \quad j = 1, 2, \dots, m, \quad k = 0, 1, \dots, M-1 \quad (22)$$

によって求まるので, (15), (16) の代わりに明示的に減算を含まない (15), (21), (22) によって  $LR$  変換が実行できる.  $RR$  変換を与える漸化式 (17), (18) についても同様に, 補助変数

$$F_1^{(n)} = E_1^{(n)} - E_1^{(n,0)}, \quad F_j^{(n)} = E_j^{(n)} - E_j^{(n,0)}, \quad j = 2, 3, \dots, m-1 \quad (23)$$

を導入すると, 補助変数に関する漸化式

$$F_{j+1}^{(n)} = \frac{E_{j+1}^{(n)}}{E_j^{(n+M)}} F_j^{(n)}, \quad j = 1, 2, \dots, m-2 \quad (24)$$

が得られるので,  $E_j^{(n+M)}$  を求めるための漸化式は

$$E_j^{(n+M)} = E_j^{(n,M)} + F_j^{(n)}, \quad j = 1, 2, \dots, m-1 \quad (25)$$

---

**Algorithm 1** differential 型のシフト付き dhToda アルゴリズム
 

---

```

1: for  $n = 0, M, 2M, \dots, \ell_{\max}$  do
2:   Choose  $s^{(n)}$  such that  $s^{(n)} < \lambda_m^{(n)}$ 
3:    $E_1^{(n,0)} = E_1^{(n)}(Q_1^{(n)}Q_1^{(n+1)} \dots Q_1^{(n+M-1)}) / (Q_1^{(n)}Q_1^{(n+1)} \dots Q_1^{(n+M-1)} - s^{(n)})$ 
4:   for  $k = 0, 1, \dots, M-1$  do
5:      $D_1^{(n+k)} = Q_1^{(n+k)}$ 
6:   end for
7:    $F_1^{(n)} = E_1^{(n)} - E_1^{(n,0)}$ 
8:   for  $j = 1, 2, \dots, m-2$  do
9:     for  $k = 0, 1, \dots, M-1$  do
10:       $Q_j^{(n+M+k)} = D_j^{(n+k)} + E_j^{(n,k)}$ 
11:       $E_j^{(n,k+1)} = Q_{j+1}^{(n+k)} E_j^{(n,k)} / Q_j^{(n+M+k)}$ 
12:       $D_{j+1}^{(n+k)} = D_j^{(n+k)} Q_{j+1}^{(n+k)} / Q_j^{(n+M+k)}$ 
13:    end for
14:     $E_j^{(n+M)} = E_j^{(n,M)} + F_j^{(n)}$ 
15:     $E_{j+1}^{(n,0)} = E_{j+1}^{(n)} E_j^{(n,M)} / E_j^{(n+M)}$ 
16:     $F_{j+1}^{(n)} = E_{j+1}^{(n)} F_j^{(n)} / E_j^{(n+M)}$ 
17:  end for
18:  for  $k = 0, 1, \dots, M-1$  do
19:     $Q_{m-1}^{(n+M+k)} = D_{m-1}^{(n+k)} + E_{m-1}^{(n,k)}$ 
20:     $E_{m-1}^{(n,k+1)} = E_{m-1}^{(n,k)} Q_m^{(n+k)} / Q_{m-1}^{(n+M+k)}$ 
21:     $D_m^{(n+k)} = D_{m-1}^{(n+k)} Q_m^{(n+k)} / Q_{m-1}^{(n+M+k)}$ 
22:  end for
23:   $E_{m-1}^{(n+M)} = F_{m-1}^{(n)} + E_{m-1}^{(n,M)}$ 
24:  for  $k = 0, 1, \dots, M-1$  do
25:     $Q_m^{(n+M+k)} = D_m^{(n+k)}$ 
26:  end for
27: end for
28: for  $k = 1, 2, \dots, m$  do
29:    $\hat{\lambda}_j = Q_j^{(\ell_{\max})} Q_j^{(\ell_{\max}+1)} \dots Q_j^{(\ell_{\max}+M-1)}$ 
30: end for

```

---

と書き換えられる。よって、(17), (18) による  $RR$  変換も (18), (23), (25) のように明示的な減算を含まない漸化式によって実行できる。まとめると、differential 型のシフト付き dhToda アルゴリズムは Algorithm 1 となる。

文献 [6] において、シフト量  $s^{(n)}$  が  $A^{(n)}$  の最小固有値未満ならば、シフト付き dhToda アルゴリズムは破綻なく実行でき、 $Q_j^{(n)}, E_j^{(n)}$  は  $n \rightarrow \infty$  においてシフトなしの場合と同じ漸近挙動 (5), (6) をもつ。differential 型のシフト付き dhToda アルゴリズムの収束速度の向上については 5 節の数値例を確認されたい。

## 4 誤差解析

本節では、文献 [7] で行われた differential 型のシフト付き dhToda アルゴリズムに対する誤差解析について述べる。dqds アルゴリズムに対する混合型誤差解析 [1] に倣って、まずは differential 型のシフト

$$\begin{array}{ccc}
L^{(n+k)}, R^{(n,k)} & \xrightarrow{\psi} & L^{(n+M+k)}, R^{(n,k+1)} \\
\psi_1 \downarrow & & \uparrow \psi_2 \\
\vec{L}^{(n+k)}, \vec{R}^{(n,k)} & \xrightarrow{\psi_{\text{exact}}} & \vec{L}^{(n+M+k)}, \vec{R}^{(n,k+1)}
\end{array}
\qquad
\begin{array}{ccc}
R^{(n,M)}, R^{(n)}, E_1^{(n,0)} & \xrightarrow{\psi'} & R^{(n+M)}, R^{(n,0)} \\
\psi'_1 \downarrow & & \uparrow \psi'_2 \\
\vec{R}^{(n,M)}, \vec{R}^{(n)}, \vec{E}_1^{(n,0)} & \xrightarrow{\psi'_{\text{exact}}} & \vec{R}^{(n+M)}, \vec{R}^{(n,0)}
\end{array}$$

図 1: 浮動小数点数演算における  $LR$  変換 (左) と  $RR$  変換 (右) の可換図.

付き dhToda アルゴリズムの 1 ステップで発生する相対誤差を見積もる. 続いて, 行列の固有値に関する相対的摂動定理 [11] を利用して, 浮動小数点数演算において  $A^{(n+M)}$  の固有値に含みうる相対誤差を見積もる.

浮動小数点数演算において, 2 つの浮動小数点数  $x, y$  に対する四則演算  $fl(x \circ y)$ ,  $\circ \in \{+, -, \times, /\}$  は

$$fl(x \circ y) = (x \circ y)(1 + \eta_1) \quad (26)$$

$$= (x \circ y)/(1 + \eta_2) \quad (27)$$

と表現できる. ただし,  $\mathbf{u}$  を計算機環境によって定まる最下位の桁の単位とすると  $|\eta_1|, |\eta_2| \leq \mathbf{u}$  である. differential 型のシフト付き dhToda アルゴリズムに対する誤差解析のために有用な 2 つの補題と 1 つの定理を導入する.

**補題 1 (Higham [8])**  $i = 1, 2, \dots, k$  に対して,  $|\delta_i| \leq \delta \ll 1$ ,  $\rho_i = \pm 1$  とする. このとき, 以下の不等式が成り立つ.

$$\left| \prod_{i=1}^k (1 + \delta_i)^{\rho_i} - 1 \right| \leq \frac{k\delta}{1 - k\delta}. \quad (28)$$

**補題 2 (Koev [11])**  $i = 1, 2$  に対して,  $|\delta_i| \leq m_i \mathbf{u}/(1 - m_i \mathbf{u})$  とする. このとき, 以下の不等式が成り立つ.

$$|(1 + \delta_1)(1 + \delta_2) - 1| \leq \frac{(m_1 + m_2)\delta}{1 - (m_1 + m_2)\delta}. \quad (29)$$

**定理 3 (Koev [11])** 各成分が非負である 2 重対角行列  $B_1, B_2, \dots, B_s$  に対して,  $A = B_1 B_2 \cdots B_s$  とする.  $x$  を  $B_1, B_2, \dots, B_s$  のある 1 つの成分,  $x$  を  $\hat{x} = x(1 + \delta)$ ,  $|\delta| \ll 1$  で置き換えた行列を  $\hat{A}$  とする. また,  $A$  の固有値  $\lambda_1, \lambda_2, \dots, \lambda_m$  を  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ ,  $\hat{A}$  の固有値  $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_m$  を  $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_m$  とすると, 以下の不等式が成り立つ.

$$|\hat{\lambda}_k - \lambda_k| \leq \frac{2|\delta|}{1 - 2|\delta|} \lambda_k, \quad k = 1, 2, \dots, m. \quad (30)$$

3 節に示したように,  $A^{(n)}$  から  $A^{(n+M)}$  が求まる differential 型のシフト付き dhToda アルゴリズムの 1 ステップでは,  $M$  回の  $LR$  変換と 1 回の  $RR$  変換が実行される. よって, まずは  $LR$  変換と  $RR$  変換の混合型誤差解析を別々に行えばよい. 浮動小数点数演算における  $LR$  変換と  $RR$  変換の混合型誤差解

析を行う際、図1のような可換図を考えるとよい。浮動小数点数の集合を  $\mathbf{F}$ 、浮動小数点数演算における  $LR$  変換を  $\psi: \mathbf{F} \rightarrow \mathbf{F}$ 、 $RR$  変換を  $\psi': \mathbf{F} \rightarrow \mathbf{F}$  とする。入力が  $\vec{L}^{(n+k)}$ ,  $\vec{R}^{(n,k)}$ 、出力が  $\check{L}^{(n+M+k)}$ ,  $\check{R}^{(n,k+1)}$  の正確な  $LR$  変換を  $\psi_{\text{exact}}$ 、入力が  $\vec{R}^{(n,M)}$ ,  $\vec{R}^{(n)}$ ,  $\vec{E}_1^{(n,0)}$ 、出力が  $\check{R}^{(n+M)}$ ,  $\check{R}^{(n,0)}$  の正確な  $RR$  変換を  $\psi'_{\text{exact}}$  とする。さらに、入力が  $L^{(n+k)}$ ,  $R^{(n,k)}$ 、出力が  $\vec{L}^{(n+k)}$ ,  $\vec{R}^{(n,k)}$  である変換を  $\psi_1$ 、入力が  $\check{L}^{(n+M+k)}$ ,  $\check{R}^{(n,k+1)}$ 、出力が  $L^{(n+M+k)}$ ,  $R^{(n,k+1)}$  である変換を  $\psi_2$  とすると、浮動小数点数演算における  $LR$  変換は  $\psi = \psi_2 \circ \psi_{\text{exact}} \circ \psi_1$  のような合成変換と解釈できる。同様に、適切な  $\psi'_1, \psi'_2$  を導入すると、浮動小数点数演算における  $RR$  変換も  $\psi' = \psi'_2 \circ \psi'_{\text{exact}} \circ \psi'_1$  のように合成変換とみなすことができる。具体的に  $\psi_1, \psi_2$  および  $\psi'_1, \psi'_2$  がどのような変換になるか調べると以下の補題が得られる。

**補題 4** 浮動小数点数演算における  $LR$  変換の入力  $L^{(n+k)}$ ,  $R^{(n,k)}$  および出力  $L^{(n+M+k)}$ ,  $R^{(n,k+1)}$  は、それぞれ  $\vec{L}^{(n+k)}$ ,  $\vec{R}^{(n,k)}$  および  $\check{L}^{(n+M+k)}$ ,  $\check{R}^{(n,k+1)}$  に比べて、各成分が相対的に高々 2ulps, 1ulp および 2ulps, 1ulp 異なる。

**補題 5** 浮動小数点数演算における  $RR$  変換の入力  $R^{(n,M)}$ ,  $R^{(n)}$  および出力  $R^{(n+M)}$ ,  $R^{(n,0)}$  は、それぞれ  $\vec{R}^{(n,M)}$ ,  $\vec{R}^{(n)}$  および  $\check{R}^{(n+M)}$ ,  $\check{R}^{(n,0)}$  に比べて、各成分が相対的に高々 1ulp, 2ulps および 2ulps, 3ulps 異なる。

補題 4, 5 を補題 1, 2 および定理 3 と組み合わせると、differential 型のシフト付き dhToda アルゴリズムの 1 ステップによって、 $A^{(n)}$  と  $A^{(n+M)}$  の固有値がどれだけ異なるかを示した定理が得られる。

**定理 6**  $A^{(n)} = L^{(n)} L^{(n+1)} \dots L^{(n+M-1)} R^{(n)}$  の固有値を  $\lambda_1, \lambda_2, \dots, \lambda_m$ 、 $A^{(n+M)} = L^{(n+M)} L^{(n+M+1)} \dots L^{(n+2M-1)} R^{(n+M)}$  の固有値を  $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_m$  とする。このとき、以下の不等式が成り立つ。

$$\frac{|\hat{\lambda}_k - \lambda_k|}{\lambda_k} \leq \frac{[(12m-4)M + 16m - 26]u}{1 - [(12m-4)M + 16m - 26]u} \quad (31)$$

定理 6 より、TN 行列の帯幅  $M$  が小さくなるほど、高精度な固有値が求まることが期待される。

## 5 数値例

本節では、3 節で導入した原点シフトによる収束速度の向上、4 節で示した誤差解析の結果を確認するための数値例を与える。

使用した計算機環境は CPU: Intel Xeon 2.27GHz, RAM: 6GB である。differential 型のシフト付き dhToda アルゴリズムは Mathematica 8.0 の倍精度演算で実行した。テスト行列は  $50 \times 50$  行列

$$A^{(0)} = L^{(0)} L^{(1)} \dots L^{(4)} R^{(0)}$$

$$L^{(i)} = \begin{pmatrix} 2 & & & & \\ 1 & 2 & & & \\ & \ddots & \ddots & & \\ & & & 1 & 2 \end{pmatrix}, \quad i = 0, 1, \dots, 4, \quad R^{(0)} = \begin{pmatrix} 1 & 1 & & & \\ & 1 & \ddots & & \\ & & \ddots & 1 & \\ & & & \ddots & 1 \\ & & & & 1 \end{pmatrix}$$

とした。真の固有値は、Mathematica 関数 Eigenvalues[ ] を数式処理モードで実行した後、倍精度に丸めた値を採用した。また、シフト量  $s^{(n)}$  は予め求められた  $A^{(0)}$  の最小固有値  $\lambda_{\min}$  を利用して、4 種類のシフト戦略  $s^{(n)} = 0, 0.5\lambda_{\min}, 0.7\lambda_{\min}, 0.9\lambda_{\min}$  を考えた。

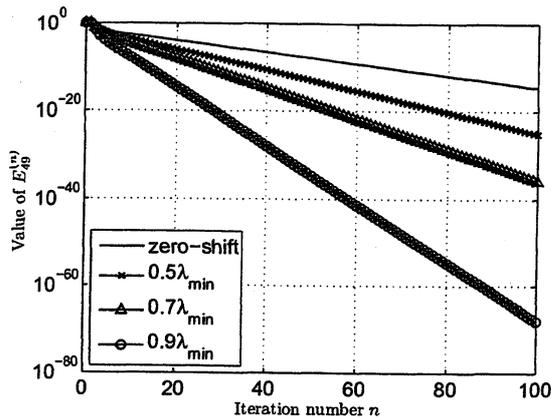


図 2:  $E_{49}^{(n)}$  の収束履歴.

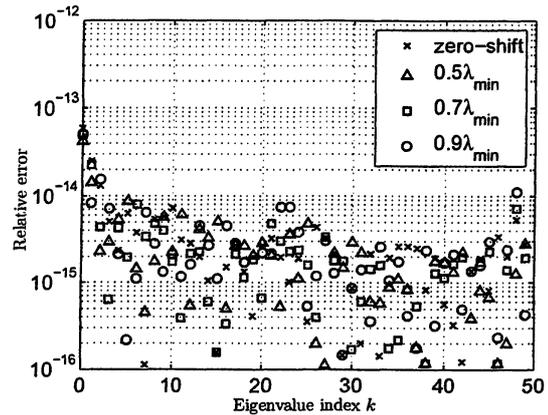


図 3: 相対誤差  $|\hat{\lambda}_k - \lambda_k| / \hat{\lambda}_k$ .

まず,  $n = 0, 1, \dots, 100$  のときの  $E_{49}^{(n)}$  の値をプロットしたグラフを図 2 に示す. シフト戦略によらず,  $n$  が大きくなるにつれて  $E_{49}^{(n)}$  は 0 に収束する様子が確認できる. また, シフト量  $s^{(n)}$  が最小固有値  $\lambda_{\min}$  に近づくにつれて, 収束速度の向上が見られる. 他のアルゴリズム変数についても, 同様にシフト導入による収束加速が観測できる. 続いて, 求められた固有値に含まれる相対誤差  $|\hat{\lambda}_k - \lambda_k| / \hat{\lambda}_k$  を図 3 に示す. ただし,  $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_m$  は真の固有値,  $\lambda_1, \lambda_2, \dots, \lambda_m$  はアルゴリズムで求めた固有値とする. シフト戦略によらず, 高精度に固有値が求まることが確認できる. これは, differential 型のシフト付き dhToda アルゴリズムが高精度という 4 節の結果に反さない.

## 6 まとめ

本論文では, TN 行列向の固有値を求めるための dhToda アルゴリズムについて 2 節で述べた. 可積分な離散ハングリー戸田方程式に基づいて定式化された dhToda アルゴリズムは, 補助変数を導入することで減算を含まない differential 型に書き換えられる. 3 節では, 原点シフトを導入した differential 型のシフト付き dhToda アルゴリズムについて述べた. differential 型のシフト付き dhToda アルゴリズムは, 既存のアルゴリズムには見られない複数の漸化式の巧みな組み合わせがポイントである. 4 節では, differential 型のシフト付き dhToda アルゴリズムに対して誤差解析を行い, アルゴリズムの 1 ステップにおける丸め誤差の影響を調べた. 浮動小数点数演算において固有値に与えられる相対的な摂動は非常に小さいことから, differential 型のシフト付き dhToda アルゴリズムは高い相対精度で固有値が求められると結論付けられる. 5 節では, 原点シフト導入の効果と誤差解析の結果を数値的に確認できた.

効率的にシフト量を見積もる方法の導出, シフト付きアルゴリズムの収束回数に関する解析は今後の課題である.

## 参考文献

- [1] K. V. Fernando, B. N. Parlett, Accurate singular values and differential qd algorithms, Numer. Math., 25 (1994), 191–229.

- [2] A. Fukuda, E. Ishiwata, M. Iwasaki, Y. Nakamura, The discrete hungry Lotka-Volterra system and a new algorithm for computing matrix eigenvalues, *Inverse Probl.*, **25** (2009), 015007.
- [3] A. Fukuda, Y. Yamamoto, M. Iwasaki, E. Ishiwata, Y. Nakamura, A Bäcklund transformation between two integrable discrete hungry systems, *Phys. Lett. A*, **375** (2011), 303–308.
- [4] A. Fukuda, A note on eigenvalue computation for a tridiagonal matrix with real eigenvalues, *J. Math. Indust.*, **3** (2011), 47–52.
- [5] A. Fukuda, E. Ishiwata, Y. Yamamoto, M. Iwasaki, Y. Nakamura, Integrable discrete hungry systems and their related matrix eigenvalues, to appear in *Ann. Mat. Pura Appl.*, DOI: 10.1007/s10231-011-0231-0.
- [6] A. Fukuda, Y. Yamamoto, M. Iwasaki, E. Ishiwata, Y. Nakamura, On a shifted  $LR$  transformation derived from the discrete hungry Toda equation, submitted.
- [7] A. Fukuda, Y. Yamamoto, M. Iwasaki, E. Ishiwata, Y. Nakamura, Error analysis for matrix eigenvalue algorithm based on the discrete hungry Toda equation, submitted.
- [8] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
- [9] M. Iwasaki, Y. Nakamura, On the convergence of a solution of the discrete Lotka-Volterra system, *Inverse Probl.*, **18** (2002), 1569–1578.
- [10] M. Iwasaki, Y. Nakamura, Positivity of dLV and mdLVs algorithms for computing singular values, *Elec. Trans. Numer. Anal.*, **38** (2011), 184–201.
- [11] P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices, *SIAM J. Matrix Anal. Appl.*, **27** (2005), 1–23.
- [12] T. Tokihiro, A. Nagai, J. Satsuma, Proof of solitonical nature of box and ball systems by means of inverse ultra-discretization, *Inverse Probl.*, **15** (1999), 1639–1662.
- [13] Y. Yamamoto, T. Fukaya, Differential qd algorithm for totally nonnegative band matrices: convergence properties and error analysis, *JSIAM Letters*, **1** (2009), 56–59.