

は、第 (i_k, j_k) 成分、 $-\sin(\theta_k)$ は、第 (j_k, i_k) 成分に、それぞれ存在する。

$$G_k(i_k, j_k) \begin{pmatrix} \vdots \\ x_{i_k} \\ \vdots \\ y_{j_k} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \sqrt{x_{i_k}^2 + y_{j_k}^2} \\ \vdots \\ 0 \\ \vdots \end{pmatrix}, \quad (2)$$

$$\cos(\theta_k) = \frac{x_{i_k}}{\sqrt{x_{i_k}^2 + y_{j_k}^2}}, \sin(\theta_k) = \frac{y_{j_k}}{\sqrt{x_{i_k}^2 + y_{j_k}^2}}. \quad (3)$$

ここで、 $\cos(\theta_k)$ 、 $\sin(\theta_k)$ を計算する際には、LAPACK の xLARTG を用いる。この変換を用いると、行列の中に任意の2つの行または列の特定の1つの成分を0に変換できる。この変換を用いると、行列 A の左から Givens 回転を作用させることで、上三角行列 R を手に入れることができる。 R を得るまでに使った Givens 回転を逆の順番で合成することにより、直交行列 Q を手に入れることができる。特定の1つの成分を0にする戦略には、さまざまな選択肢があり、代表的な戦略として、

$$\begin{aligned} & G(n, n+1) \cdots G(n, m-1)G(n, m) \\ & \times \cdots \\ & \times G(2, 3) \cdots G(2, m-1)G(2, m) \\ & \times G(1, 2) \cdots G(1, m-1)G(1, m)A = R. \end{aligned} \quad (4)$$

がある。最下段の $G(1, 2), \dots, G(1, m-1), G(1, m)$ の値は、第1列目の非対角成分を0にすることを目的として計算し、最下段から1つ上の $G(2, 3), \dots, G(2, m-1), G(2, m)$ の値は、第2列目の非対角成分を0にすることを目的として計算し、最上段の $G(n, n+1), \dots, G(n, m-1), G(n, m)$ の値は、第 n 列目の非対角成分を0にすることを目的として計算する。この戦略では、 Q は次のように計算できる。

$$\begin{aligned} Q = & G(1, m)^\top G(1, m-1)^\top \cdots G(1, 2)^\top \\ & \times G(2, m)^\top G(2, m-1)^\top \cdots G(2, 3)^\top \\ & \times \cdots \\ & \times G(n, m)^\top G(n, m-1)^\top \cdots G(n, n+1)^\top. \end{aligned} \quad (5)$$

なお、ここでは、添え字の k の部分については、省略している。ここで、TRL 法における、収束の終盤に現れる S_m について考える。収束の終盤に現れる S_m は、対角成分以外の成分の値が絶対値の意味で非常に小さい行列である。 S_m に対する固有値分解は、

$$S_m Y_m = Y_m D_m \quad (6)$$

と表せる。ここで、 Y_m は、固有ベクトルを並べた行列である。特に、TRL 法においては、収束の終盤に現れる S_m は対

角行列に近づく。さらに、Extract the required eigenvalues $\gamma_1, \dots, \gamma_l$ and the eigenvectors $\mathbf{y}_1, \dots, \mathbf{y}_l$ のために、 Y の順序を適切に並び替える操作も行われる。すると、収束の終盤では、 Y も対角成分に近い行列になる。この固有ベクトルを並べた行列 Y に対する QR 分解の戦略は、収束の終盤では、 Y が対角成分に近い行列になることを考慮すると、式 (4) の戦略ではなく、

$$\begin{aligned} & G(n, n+1) \cdots G(m-2, m-1)G(m-1, m) \\ & \times \cdots \\ & \times G(2, 3) \cdots G(m-2, m-1)G(m-1, m) \\ & \times G(1, 2) \cdots G(m-2, m-1)G(m-1, m)A = R. \end{aligned} \quad (7)$$

とする。最下段の $G(1, 2), \dots, G(m-2, m-1), G(m-1, m)$ の値は、第1列目の非対角成分を0にすることを目的として計算し、最下段から1つ上の $G(2, 3), \dots, G(m-2, m-1), G(m-1, m)$ の値は、第2列目の非対角成分を0にすることを目的として計算し、最上段の $G(n, n+1), \dots, G(m-2, m-1), G(m-1, m)$ の値は、第 n 列目の非対角成分を0にすることを目的として計算する。この戦略では、 Q は次のように計算できる。

$$\begin{aligned} Q = & G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(1, 2)^\top \\ & \times G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(2, 3)^\top \\ & \times \cdots \\ & \times G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(n, n+1)^\top. \end{aligned} \quad (8)$$

TRL 法における固有ベクトルを並べた行列 Y_l として、 $Y_l := [\mathbf{y}_1 \ \dots \ \mathbf{y}_l]$ ではなく、Givens 回転を並べた式 (8) の Q を採用できる。

計算機上で Givens 回転を実行する場合、丸め誤差の影響により、オーバーフローやアンダーフローが生じることがある。これを避けるために、Algorithm 1 のように Givens 回転を実装すべきである。融合積和演算は、Algorithm 1 の12行目と17行目の二重下線部分で採用することができる。TRL 法が実装されている TRLAN に対して、丸め誤差の影響を避けるために、本稿では、Algorithm 1 を適用し、これを従来法とする。

3. 新しいリスタート戦略

3.1 固有値分解におけるレイリー商

固有値分解におけるレイリー商 [6] を、

$$\rho = \frac{1}{\|\tilde{\mathbf{x}}_i\|^2} \tilde{\mathbf{x}}_i^\top A \tilde{\mathbf{x}}_i. \quad (9)$$

と定義する。式 (9) の ρ は、計算された固有ベクトル $\tilde{\mathbf{x}}_i$ とする場合、

$$\rho = \arg \min_z \|A \tilde{\mathbf{x}}_i - z \tilde{\mathbf{x}}_i\|^2. \quad (10)$$

Algorithm 1 Givens 回転の実装方法

```

1:  $f \leftarrow |x_{ik}|$ 
2:  $g \leftarrow |y_{jk}|$ 
3:  $t \leftarrow \max(f, g)$ 
4: if  $t = 0$  then
5:    $\cos(\theta_k) \leftarrow 1$ 
6:    $\sin(\theta_k) \leftarrow 0$ 
7:    $\sqrt{x_{ik}^2 + y_{jk}^2} \leftarrow 0$ 
8: else
9:    $u \leftarrow f/t$ 
10:   $v \leftarrow g/t$ 
11:  if  $f \geq g$  then
12:     $r \leftarrow \sqrt{1+v^2}$ 
13:     $\cos(\theta_k) \leftarrow u/r$ 
14:     $\sin(\theta_k) \leftarrow v/r$ 
15:     $\sqrt{x_{ik}^2 + y_{jk}^2} \leftarrow r \times t$ 
16:  else
17:     $r \leftarrow \sqrt{1+u^2}$ 
18:     $\cos(\theta_k) \leftarrow u/r$ 
19:     $\sin(\theta_k) \leftarrow v/r$ 
20:     $\sqrt{x_{ik}^2 + y_{jk}^2} \leftarrow r \times t$ 
21:  end if
22: end if

```

を満たす。この際、 ρ は、 A の固有値 λ_i ($i = 1, \dots, n$) に非常に近い値となる。

3.2 実装

TRL 法では、小さい行列 S_m の EVD が内部的に実行され、その結果を用いて、リスタートが行われる。計算誤差を考慮しない限り、EVD によって得られる固有ベクトルは直交行列である。しかし、丸め誤差のために、Lanczos 法 [4] の直交性が悪化することが知られている。この問題を回避するために、小行列 S_m の固有ベクトルの直交化を用いてリスタートする方法が提案されている。本稿では、Givens 回転を用いた QR 分解 [5] によって、固有ベクトルを列直交行列と上三角行列に分解することにより、最大直交性を持つ S_m の固有ベクトルを計算させる。アルゴリズム 2 は、その実装方法である。

従来の方法では、 l ベクトルが固有ベクトル $\mathbf{y}_1, \dots, \mathbf{y}_m$ から Y_l として抽出される。ここで、 Y_l は、 $m \times l$ の行列である。新しい戦略では、 Y_l を直交化するために、 $Y_l = QR$ の Givens 回転を用いた QR 分解を用いる。直交行列 Q を新しい Y_l とすると

$$Y_l \leftarrow [y_1, y_2, \dots, y_l], \quad (11)$$

$$Y_l = QR, \quad (12)$$

$$Y_l \leftarrow Q. \quad (13)$$

と表される。

$$S_m Y_l = Y_l D_l, \quad (14)$$

を満たすために、

Algorithm 2 新しいリスタート戦略による TRL 法の実装

```

1: Set  $m$  and  $l$ ;
2: Input Lanczos decomposition  $AV_m = V_m S_m + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^\top$ ;
3: for  $i := 1, 2, \dots$  do
4:   Compute all eigenvalue  $\gamma_1, \dots, \gamma_m$  and the normalized
     eigen vectors  $\mathbf{y}_1, \dots, \mathbf{y}_m$  corresponding to the eigenvalue
     in  $S_m$ ;
5:   Extract the required eigenvalues  $\gamma_1, \dots, \gamma_l$  and the eigen
     vectors  $\mathbf{y}_1, \dots, \mathbf{y}_l$ ;
6:    $D_l := \text{diag}(\gamma_1, \dots, \gamma_l)$ ;
7:    $Y_l := [\mathbf{y}_1 \dots \mathbf{y}_l]$ ;
8:   Compute the QR Decomposition using Givens rotation of
      $Y_l = QR$ 
9:    $Y_l \leftarrow Q$ 
10:   $[D_l]_{i,i} \leftarrow [Y_l^\top S_m Y_l]_{i,i}$  for  $i = 1, \dots, l$ 
11:   $\tilde{V}_l := V_m Y_l$ ;
12:   $\boldsymbol{\eta} := \mathbf{e}_m^\top Y_l$ ;
13:   $\tilde{\mathbf{v}}_{l+1} := \mathbf{v}_{m+1}$ ;
14:   $\tilde{\mathbf{r}}_{l+1} := A \tilde{\mathbf{v}}_{l+1}$ ;
15:   $\tilde{\alpha}_{l+1} := \tilde{\mathbf{v}}_{l+1}^\top \tilde{\mathbf{r}}_{l+1}$ ;
16:   $\tilde{\mathbf{r}}_{l+1} := \tilde{\mathbf{r}}_{l+1} - \sum_{j=1}^l \beta_m \boldsymbol{\eta}(j) \tilde{\mathbf{v}}_j - \tilde{\alpha}_{l+1} \tilde{\mathbf{v}}_{l+1}$ ;
17:   $\tilde{\beta}_{l+1} := |\tilde{\mathbf{r}}_{l+1}|$ ;
18:   $\tilde{\mathbf{v}}_{l+2} := \tilde{\mathbf{r}}_{l+1} / \tilde{\beta}_{l+1}$ ;
19:   $\tilde{V}_{l+1} := [\tilde{V}_l \ \tilde{\mathbf{v}}_{l+1}]$ ;
20:   $\tilde{S}_{l+1} := \begin{bmatrix} D_l & \beta_m \boldsymbol{\eta}^\top \\ \beta_m \boldsymbol{\eta} & \tilde{\alpha}_{l+1} \end{bmatrix}$ ;
21:  Adopt  $A \tilde{V}_{l+1} = \tilde{V}_{l+1} \tilde{S}_{l+1} + \tilde{\beta}_{l+1} \tilde{\mathbf{v}}_{l+2} \mathbf{e}_{l+1}^\top$  to the Lanczos
     method at  $m - l - 1$  times;
22: end for

```

$$[D_l]_{i,i} \leftarrow [Y_l^\top S_m Y_l]_{i,i}. \quad (15)$$

に近似させる。ここで、 $[D_l]_{i,i}$ のレイリー商であり、固有ベクトル Y_l を用いて S_m を厳密に近似する。直交性が改善された式 (13) を用いて、

$$x = \|S_m Y_l - Y_l D_l\|, \quad (16)$$

を、アルゴリズム 2 の 10 行目で計算された固有値 $D_l(1:l, 1:l)$ を用いて定義する。 Y_l の直交性を改善することによって、 x が大きくなる。この問題を避けるために、 D_l を式 (15) のレイリー商を用いて、再定義する。さらに、

$$\begin{aligned} AV_m Y_l &= V_m S_m Y_l + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^\top Y_l \\ &= V_m Y_l D_l + \beta_m \mathbf{v}_{m+1} \mathbf{e}_m^\top Y_l. \end{aligned} \quad (17)$$

より、

$$V_l^\top AV_l = Y_l^\top S_m Y_l = D_l, \quad (18)$$

が導き出される。ゆえに、 A の固有値に近い Y_l , $[D_l]_{i,i}$ は、レイリー商として扱うことができる。

$$\tilde{V}_l := V_m Y_l, \quad (19)$$

$$\boldsymbol{\eta} := \mathbf{e}_m^\top Y_l, \quad (20)$$

を用いて、新しい戦略を用いた TRL 法は、

$$\begin{aligned} \tilde{V}_l &:= V_m G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(1, 2)^\top \\ &\quad \times G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(2, 3)^\top \\ &\quad \times \cdots \\ &\quad G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(l, l+1)^\top. \end{aligned} \quad (21)$$

によって実装される. 式 (21) は, LAPACK の `xROT` を用いて計算される. 式 (8) を用いて新しい戦略を用いた TRL 法の 10 行目は, 以下のように実装される.

$$\begin{aligned} D_l &\leftarrow G(l, l+1) \cdots G(m-2, m-1) G(m-1, m) \\ &\quad \times \cdots \\ &\quad \times G(2, 3) \cdots G(m-2, m-1) G(m-1, m) \\ &\quad \times G(1, 2) \cdots G(m-2, m-1) G(m-1, m) \\ &\quad \times S_m G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(1, 2)^\top \\ &\quad \times G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(2, 3)^\top \\ &\quad \times \cdots \\ &\quad \times G(m-1, m)^\top G(m-2, m-1)^\top \cdots G(l, l+1)^\top. \end{aligned} \quad (22)$$

さらに,

$$X^{(0)} := S_m, \quad (23)$$

$$X^{(1)} := G(m-1, m) X^{(0)} G(m-1, m)^\top, \quad (24)$$

$$X^{(2)} := G(m-2, m-1) X^{(1)} G(m-2, m-1)^\top, \cdots \quad (25)$$

となる. 式 (23), (24), (25) は, LAPACK の `xROT` を用いて計算することができる.

$X^{(i)}$ において, $X_{m-1, m}^{(i)}$, $X_{m, m-1}^{(i)}$, $X_{m-1, m-1}^{(i)}$, $X_{m, m}^{(i)}$ を除くすべての要素に関して, 単純に `xROT` を用いて実装できる. 数学では, $X^{(i)}$ は常に対称行列である. したがって, 実装の際には, 対称性を維持させなければならない. 加えて, 対角要素は高い精度で計算されなければならない. 以上の要求を考慮し,

$$pp = X_{m-1, m-1}^{(i)}, \quad (26)$$

$$pq = X_{m-1, m}^{(i)}, \quad (27)$$

$$qq = X_{m, m}^{(i)}, \quad (28)$$

$$\begin{aligned} X_{m-1, m}^{(i)} &= \cos(\theta_k) \times \sin(\theta_k) \times (qq - pp) \\ &\quad + pq \times (\cos(\theta_k) - \sin(\theta_k)) \\ &\quad \times (\cos(\theta_k) + \sin(\theta_k)), \end{aligned} \quad (29)$$

$$X_{m, m-1}^{(i)} = X_{m-1, m}^{(i)}, \quad (30)$$

$$\begin{aligned} X_{m-1, m-1}^{(i)} &= \cos(\theta_k) \times \cos(\theta_k) \times pp \\ &\quad + 2 \cos(\theta_k) \times \sin(\theta_k) \times pq \\ &\quad + \sin(\theta_k) \times \sin(\theta_k) \times qq, \end{aligned} \quad (31)$$

$$\begin{aligned} X_{m, m}^{(i)} &= \sin(\theta_k) \times \sin(\theta_k) \times pp \\ &\quad - 2 \cos(\theta_k) \times \sin(\theta_k) \times pq \\ &\quad + \cos(\theta_k) \times \cos(\theta_k) \times qq, \end{aligned} \quad (32)$$

を用いる.

4. まとめ

本稿では, 大規模スパース行列の部分 EVD を計算するための TRL 法の改良法を示している. 改良法では, TRL 法の内部で生成された小行列 S_m の固有ベクトルの直交化を用いてリスタートを行う. リスタートの際に, 改良法を用いた実装では, Givens 回転に基づいて QR 分解を行う.

単精度浮動小数点演算における性能を数値実験によって示している. この際, TRLAN に実装されている TRL 法から丸め誤差の影響をなくすために, 本稿で示した Givens 回転の実装方法を適用している. この適用された従来法と新しいリスタート戦略との比較実験を行い, 反復回数と計算時間が減少することを確認している.

謝辞 本研究は JSPS 科研費 JP17H02858 と JP17H00167 の助成を受けたものです.

参考文献

- [1] Baglama, J., and Reichel, L.: *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM Journal on Scientific Computing, 27(1), pp.19–42 (2005).
- [2] Calvetti, D., et al. *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electronic Transactions on Numerical Analysis, 2(1), pp.1–21 (1994).
- [3] Ishida, Y., Takata, M., Kimura, K., and Nakamura Y.: *Improvement of the Augmented Implicitly Restarted Lanczos Bidiagonalization Method in Single Precision Floating Point Arithmetic*, IPSJ Transactions on Mathematical Modeling and its Applications, 11(3), pp.19–25 (2018).
- [4] Lanczos, C.: *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bureau Standards, Sec, vol.B, no.45, pp.255–282(1950).
- [5] Golub, G. H., and Van Loan, C. F.: *Matrix Computations.*, Johns Hopkins University Press, 4th edition (2012).
- [6] Parlett, B. N.: *The Symmetric Eigenvalue Problem*, Society for Industrial and Applied Mathematics (1998).
- [7] Wu, K., and Simon, H.: *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. and Appl., 22(2), pp.602–616(2000).
- [8] Wu, J.: *Research on Eigenvalue Computations*, <https://sdm.lbl.gov/kewu/ps/trlan-ug.pdf>, (1999).