

Supertropical 代数上の行列の特異値分解に対する組合せ論的アルゴリズム

西田 優樹¹, 古池 郁美¹, 岩崎 雅史¹

¹ 京都府立大学生命環境科学研究科

e-mail : y-nishida@kpu.ac.jp

1 概要

Max-plus 代数とは集合 $\mathbb{R}_{\max} := \mathbb{R} \cup \{-\infty\}$ 上に加法 \oplus と乗法 \otimes をそれぞれ

$$a \oplus b := \max(a, b), \quad a \otimes b := a + b$$

で定義した半環である。加法についての単位元は $\varepsilon := -\infty$ であるが、加法の逆元は存在しない。この点を解消するために、ghost とよばれる元の集合 $\mathbb{R}^\nu := \{a^\nu \mid a \in \mathbb{R}\}$ を付加した supertropical 代数が考案された [1]。ここでは、ghost はいわば「値を持った零元」であり、supertropical 代数での加法 \oplus_g と乗法 \otimes_g は

$$a \oplus_g b = \begin{cases} a & (|a| > |b|), \\ b & (|a| < |b|), \\ a^\nu & (|a| = |b|), \end{cases} \quad a \otimes_g b = \begin{cases} a + b & (a, b \in \mathbb{R}_{\max}), \\ (|a| + |b|)^\nu & (a \in \mathbb{R}^\nu \text{ または } b \in \mathbb{R}^\nu) \end{cases}$$

で定義される。ただし、 $a \in \mathbb{R}$ に対して $|a| = |a^\nu| = a$ である。また等号に代わる非対称な関係

$$a \vDash b \Leftrightarrow a = b \text{ または } (|a| \geq |b| \text{ かつ } a \in \mathbb{R}^\nu)$$

が定義される。本研究では、max-plus 行列の supertropical 代数上での特異値分解を考える。Max-plus 代数および supertropical 代数上での行列演算を通常の線形代数の場合と同様に定義することで、行列 $A \in \mathbb{R}_{\max}^{m \times n}$ の特異値分解は次の形で考えることができる。

$$\begin{cases} U \otimes_g \Sigma \otimes_g V^\top \vDash A, \\ U^\top \otimes_g U \vDash E_m, \\ V^\top \otimes_g V \vDash E_n. \end{cases} \quad (1)$$

ただし、 Σ は $i = 1, 2, \dots, \min(m, n)$ について (i, i) 成分が σ_i でそのほかの成分がすべて ε である $m \times n$ 行列であり、 E_m, E_n は max-plus 代数における単位行列で、それらの次数はそれぞれ m 次、 n 次である。第 2 式および第 3 式は特異ベクトルの max-plus 代数の意味での直交性を表す。

Max-plus 行列の特異値分解は [2] において提案されているが、その手法は逆超離散化に類似した写像を通して通常の代数系での計算に帰着させるものであり、パラメトリックな行列の特異値分解を要する。また、max-plus 代数上の特殊な固有多項式を利用した特異値の導出法が [3] で提案されているが、特異ベクトルの計算方法は記されていない。そこで本研究では、組合せ論的な方法で特異値および特異ベクトルを計算し、特異値分解を求めるアルゴリズムを提案する。

2 特異値分解アルゴリズム

以下では簡単のため $m \leq n$ とする。また、ベクトル u の第 i 成分を $[u]_i$ で、行列 A の (i, j) 成分を $[A]_{i,j}$ で表す。さらに、 $P \in \mathbb{R}_{\max}^{n \times n}$ に対して

$$P^+ := P \oplus P^{\otimes 2} \oplus \dots \oplus P^{\otimes n}$$

と定義する．このとき，行列 $A \in \mathbb{R}_{\max}^{m \times n}$ の特異値は次のアルゴリズムで与えられる．

アルゴリズム 1 入力: Max-plus 行列 $A \in \mathbb{R}_{\max}^{m \times n}$

出力: A の特異値 $\sigma_1, \sigma_2, \dots, \sigma_m$ およびベクトル $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_m$

- (i) $A^{(0)} := A$ および $k := 1$ とおく．
- (ii) $[A^{(k-1)}]_{p_k, q_k} = \max_{i,j} [A^{(k-1)}]_{i,j}$ となる組 (p_k, q_k) を求め， $\sigma_k := [A^{(k-1)}]_{p_k, q_k}$ とおく．
- (iii) 行列 $((-2\sigma_k) \otimes A^{(k-1)} \otimes (A^{(k-1)})^\top)^+$ の p_k 列目と行列 $((-2\sigma_k) \otimes (A^{(k-1)})^\top \otimes A^{(k-1)})^+$ の q_k 列目をそれぞれ u_k, v_k とおく．
- (iv) $k = m$ であればアルゴリズムを終了する．そうでなければ， $A^{(k)} := A^{(k-1)} \oplus u_k \otimes \sigma_k \otimes v_k^\top$ として，さらに $A^{(k)}$ の p_k 行目と q_k 列目の成分をすべて ε に置き換える．その後 $k := k + 1$ として (ii) に戻る．

上記のアルゴリズムで得られるベクトル u_1, u_2, \dots, u_m および v_1, v_2, \dots, v_m をそのまま並べても直交行列は得られない．そこで，以下のアルゴリズム 2 によってこれらのベクトルを直交化する．簡単のため， $\{1, 2, \dots, n\} \setminus \{q_1, q_2, \dots, q_m\} = \{q_{m+1}, q_{m+2}, \dots, q_n\}$ とする．

アルゴリズム 2 入力: アルゴリズム 1 で得られたベクトル u_1, u_2, \dots, u_m および v_1, v_2, \dots, v_m

出力: supertropical 代数の意味での直交行列 $U \in \mathbb{R}_{\max}^{m \times m}$ および $V \in \mathbb{R}_{\max}^{n \times n}$

- (i) $U^{(0)} := E_m, V^{(0)} := E_n, k := 1$ とおく．
- (ii) $U^{(k-1)}$ の p_k 列目をベクトル u_k で置き換えた行列を $U^{(k)}$ とおく．同様に， $V^{(k-1)}$ の q_k 列目をベクトル v_k で置き換えた行列を $V^{(k)}$ とおく．
- (iii) 行列 U と V の k 列目をそれぞれ

$$[U]_{p_i, k} = \begin{cases} [(U^{(k)})^\top \otimes U^{(k)}]^+_{p_i, p_k} & (i \leq k-1 \text{ のとき}) \\ [u_k]_{p_i} & (i \geq k \text{ のとき}) \end{cases}$$

$$[V]_{q_i, k} = \begin{cases} [(V^{(k)})^\top \otimes V^{(k)}]^+_{q_i, q_k} & (i \leq k-1 \text{ のとき}) \\ [v_k]_{q_i} & (i \geq k \text{ のとき}) \end{cases}$$

で定義する．

- (iv) $k = m$ であれば， V の $m+1, m+2, \dots, n$ 列目を

$$[V]_{q_i, j} = [((V^{(m)})^\top \otimes V^{(m)})^+]_{q_i, q_j} \quad (i = 1, 2, \dots, n, j = m+1, m+2, \dots, n)$$

によって定め，アルゴリズムを終了する．そうでなければ， $k := k + 1$ として (ii) に戻る．

定理 1 アルゴリズム 1 およびアルゴリズム 2 で得られる $\sigma_1, \sigma_2, \dots, \sigma_m$ と行列 U, V を用いて，行列 A は (1) の形に特異値分解できる．

参考文献

- [1] Z. Izhakian, L. Rowen, Supertropical algebra, Adv. Math., 225 (2010), 2222–2286.
- [2] B. De Schutter, B. De Moor, The QR decomposition and the singular value decomposition in the symmetrized max-plus algebra revisited, SIAM Review, 44 (2002), 417–454.
- [3] J. Hook, Max-plus singular values, Linear Algebra Appl., 456 (2015), 419–442.

実対称行列に対する部分固有対の逆反復改良法

寺尾 剛史¹, 尾崎 克久²

¹ 九州大学, ² 芝浦工業大学

e-mail : terao.takeshi.412@m.kyushu-u.ac.jp

1 概要

本論文では、実対称行列 $A = A^T \in \mathbb{R}^{n \times n}$ に対する固有値問題 $Ax = \lambda x, x \neq \mathbf{0}$ を扱う。行列 A に対するすべての固有値の集合を $\Lambda(A)$ 、その部分集合を $S(A) \subset \Lambda(A)$ と表記する。また、固有値 $\mu_i \in S(A)$ に対応する固有ベクトルを x_i と表記する。ここで、 A に対するすべての近似固有対が得られた場合、効率的な反復改良法が提案されている [1, 2]。また、計算コストを低減した混合精度計算法が提案されている [3]。

しかし、応用上では、すべての固有対の情報が必要ない場合や、その計算コストが高い場合がある。ここで、すべての近似固有対が求まっていない場合の反復改良法を考える。我々は、 $\mu_i \in S(A), \nu_i \in \Lambda(A) \setminus S(A)$ としたとき、すべての i, j に対して $|\mu_i| > |\nu_j|$ かつ $|\mu_i - \mu_j| \geq \epsilon$ の場合で、部分固有対の反復改良が可能であることを示した [4]。ここで、 ϵ は比較的小さい値のスカラーである。この手法の利点として、反復改良法で要求される主要な計算で、行列積のみに高い精度が要求されることである。

2 部分固有対に対する反復改良法

すべての i, j に対して $|\mu_i| > |\nu_j|, (1 \leq i \leq k)$ かつ $|\mu_i - \mu_j| \geq \epsilon$ を仮定する。また、

$$D = \text{diag}(\mu_1, \dots, \mu_k), \quad X = (x_1, \dots, x_k), \quad k < n \quad (1)$$

とし、それら初期近似値を $\hat{D}^{(0)}, \hat{X}^{(0)}$ とする。このときの反復改良アルゴリズムが以下である：

Algorithm 1 Refinement for an approximate eigenpair subset

Input: $A \in \mathbb{R}^{n \times n}$, $\hat{X}^{(p)} \in \mathbb{R}^{n \times k}$, and $\hat{D}^{(p)} \in \mathbb{R}^{k \times k}$ with $k \leq n$

1: $[Y, T] = \text{CompactWY}(\hat{X}^{(p)})$

2: $B = A\hat{X} - \hat{X}\hat{D}^{(p)}$

3: $P = \hat{H}^T B$, where $\hat{H} = I_n - YTY^T$

4: $q_j = \tilde{x}_j^T \tilde{x}_j$ for $1 \leq j \leq k$

5: $\hat{d}_{jj}^{(p+1)} = \hat{d}_{jj}^{(p)} + p_{jj}/q_j$ for $1 \leq j \leq k$

6: $\hat{e}_{ij} = \begin{cases} (1 - q_j)/2, & i = j \\ \frac{p_{ij}}{\hat{d}_{jj}^{(p+1)} - \hat{d}_{ii}^{(p+1)}}, & 1 \leq i, j \leq k \\ p_{ij}/\hat{d}_{jj}^{(p+1)}, & k+1 \leq i \leq n, 1 \leq j \leq k \end{cases}$

7: $\hat{X}^{(p+1)} = \hat{X} + \hat{H}\hat{E}$

Output: $\hat{X}^{(p+1)}, \hat{D}^{(p+1)}$

Algorithm 1 の 1 行目 $[Y, T] = \text{CompactWY}(X)$ は、 X に対する Compact WY 表現 [5, 6] の計算を意味する。つまり、 $H = I - YTY^T$ としたとき、 H は直交行列であり、 $H^T X$ が矩形単位行列

となる。

3 逆反復改良法

すべての i, j に対して $|\mu_i| < |\nu_j|$ が成り立つ場合を考える。このとき、 A が正則であれば、

$$A^{-1}X = XD^{-1} \quad (2)$$

であり、 $|\mu_i^{-1}| > |\nu_j^{-1}|$ が成り立つ。ここで、逆反復改良には高精度な $A^{-1}\hat{X} - XD^{-1}$ が要求される (Algorithm 1, line 2)。この式に対して

$$A^{-1}\hat{X} - XD^{-1} = A^{-1}(\hat{X} - AXD^{-1}) \quad (3)$$

と式変形することで、行列ベクトル積 $R := \hat{X} - AXD^{-1}$ が十分に高精度に計算されている場合、 $A^{-1}R$ は低精度計算でよい。これにより、予め行列 A の LU 分解の結果などが必要だがその精度は低精度でよく、反復改良で求められる高精度計算の殆どが行列積である。

参考文献

- [1] Takeshi Ogita and Kensuke Aishima. Iterative refinement for symmetric eigenvalue decomposition. *Japan Journal of Industrial and Applied Mathematics*, Vol. 35, No. 3, pp. 1007–1035, 2018.
- [2] Takeshi Ogita and Kensuke Aishima. Iterative refinement for symmetric eigenvalue decomposition II: clustered eigenvalues. *Japan Journal of Industrial and Applied Mathematics*, Vol. 36, No. 2, pp. 435–459, 2019.
- [3] 内野佑基, 尾崎克久, 荻田武史. 実対称行列の固有値分解に対する反復改良法の大規模並列環境における実装と評価. 研究報告ハイパフォーマンスコМПユーティング (HPC), Vol. 2020, No. 15, pp. 1–7, 2020.
- [4] Takeshi Terao, Imamura Toshiyuki, and Katsuhisa Ozaki. Iterative refinement for an eigenpair subset of a real symmetric matrix. JSIAM Letters, Accepfed for publication.
- [5] Yusaku Yamamoto. Aggregation of the compact WY representations generated by the TSQR algorithm. In *Conference talk presented at SIAM Applied Linear Algebra*, 2012.
- [6] Grey Ballard, James Demmel, Laura Grigori, Mathias Jacquelin, Hong Diep Nguyen, and Edgar Solomonik. Reconstructing householder vectors from tall-skinny QR. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, pp. 1159–1170. IEEE, 2014.

可変的前処理付き block CG 法の内部と外部の反復における誤差の繋がり

山本 実央¹, 相原 研輔¹

¹ 東京都市大学

e-mail : g2381458@tcu.ac.jp

1 はじめに

大規模疎行列を係数に持つ連立一次方程式 $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ を反復法によって数値的に解くことは, 科学技術計算において重要である. 近年, 複数の右辺ベクトルを持つ連立一次方程式

$$AX = B, \quad B = [b_1, b_2, \dots, b_s] \in \mathbb{R}^{n \times s}$$

を効率よく求解するためのブロック型反復法が注目されている. 特に, block CG 法 [1] は, 係数行列 A が正定値対称である場合に有効である最も代表的なブロック型反復法である. また反復法では, 収束性を改善する手法として, 前処理を併用することが有効である.

反復毎に異なる前処理を適用できる可変的前処理 [2, 3] は, 汎用性が高い手法であるが, その効果や収束性については未だ不明な点も多い. そこで本研究では, block CG 法に可変的前処理を適用した場合の収束性への影響を数理的に解析する. 具体的には, 反復行列の更新に用いる $s \times s$ 行列の計算法の違いによって, 可変的前処理付き block CG 法のアルゴリズムは複数存在することに注目する. 本研究では, 残差や探索方向について局所的な直交性を保持する前処理付きアルゴリズムを取り上げ, その収束性を考察する. 特に, 可変的前処理では, 反復毎に残差を右辺項とする線形方程式を反復法によって粗く解くアプローチがよく用いられるが (これを内部反復と呼ぶ), その内部反復が, 基盤となる block CG 法の反復 (外部反復と呼ぶ) の収束性改善に有効であることを示唆する収束定理を示す. また, この収束定理の妥当性を確認するための数値実験も示す.

2 可変的前処理付き block CG 法の収束性

可変的前処理付き block CG 法のアルゴリズムを, Algorithm 1 に示す. 一般的な前処理では, 固定の前処理行列 $M \approx A$ に対し, 反復毎に $Z_k = M^{-1}R_k$ を計算する. 一方, 可変的前処理では, 反復毎に異なる前処理行列 M_k を適用できるが, 実装上は方程式 $AZ_k = R_k$ を適当な反復法により粗

Algorithm 1 Block CG method with variable preconditioning (Bl-VPCG).

- 1: Select an initial guess X_0 , and compute $R_0 = B - AX_0$.
 - 2: Solve $AZ_0 = R_0$ roughly, and set $P_0 = Z_0$.
 - 3: **for** $k = 0, 1, 2, \dots$, until convergence **do**
 - 4: $\alpha_k^\square = (P_k^\top AP_k)^{-1}(P_k^\top R_k)$
 - 5: $X_{k+1} = X_k + P_k \alpha_k^\square$, $R_{k+1} = R_k - AP_k \alpha_k^\square$
 - 6: Solve $AZ_{k+1} = R_{k+1}$ roughly.
 - 7: $\beta_k^\square = -(P_k^\top AP_k)^{-1}(P_k^\top AZ_{k+1})$
 - 8: $P_{k+1} = Z_{k+1} + P_k \beta_k^\square$
 - 9: **end for**
-

く解くことで Z_k に対応する行列を求める場合が多い．内部反復，すなわち方程式 $AZ_k = R_k$ の粗い求解には，外部反復と同様に block CG 法などが用いられる．

さて，通常の block CG 法やその前処理付きアルゴリズムでは，計算量の削減のため， $\alpha_k^\square, \beta_k^\square$ の計算に現れる $s \times s$ 行列を再利用性のある形式に変換する．しかし，本研究ではそのような変換はあえて用いないこととする．それは，次の局所的な直交性を保持するためである．

$$R_k^\top P_{k-1} = O, \quad P_k^\top A P_{k-1} = O, \quad k = 1, 2, \dots$$

この直交性を用いることで，Algorithm 1 に対して以下の補題を得る．ただし，行列 $X \in \mathbb{R}^{n \times s}$ に対し， F_A ノルムを $\|X\|_{F_A} := \sqrt{\text{tr}(X^\top A X)}$ と定める．本研究では A を正定値対称行列としているため， F_A ノルムは行列ノルムの基本的な性質である正定値性，斉次性，劣加法性を満たすが，劣乗法性は満たさないことを付記しておく．

補題 1 行列 Z_k を Algorithm 1 の内部反復で得られる近似解とする．このとき，以下の不等式が成り立つ．ただし， $X^* := A^{-1}B$ かつ $Z_k^* := A^{-1}R_k$ とする．

$$\|X^* - X_{k+1}\|_{F_A} \leq \|Z_k^* - Z_k\|_{F_A}.$$

補題 1 より，内部反復における誤差の F_A ノルムは，次の外部反復での誤差の F_A ノルムの上限となる．また，補題 1 を用いると，可変的前処理付き block CG 法に対し，次の収束定理が得られる．

定理 2 定数 θ_k ($0 < \theta_k < 1$) に対し，Algorithm 1 の内部反復において，以下を満たす近似解 Z_k が得られたとする．

$$\|Z_k^* - Z_k\|_{F_A} \leq \theta_k \|Z_k^*\|_{F_A}.$$

このとき，外部反復の近似解 X_k について，以下が成り立つ．

$$\|X^* - X_{k+1}\|_{F_A} \leq \theta_k \|X^* - X_k\|_{F_A}.$$

定理 2 より， F_A ノルムにおいて，内部反復における相対的な許容誤差は，外部反復の一反復における誤差の減少率に対応付けられる．なお，以上の結果は，VPGCR 法に対する解析 [2, Lemma 1, Theorem 1] に基づき，VPCG 法に対する収束定理 [3, Theorem 10.2] の一部をブロック型に拡張したものである．補題・定理の証明や数値実験については，当日の発表にて示す．

謝辞 本研究の一部は科学研究費補助金（基盤研究 (C)：21K11925）の助成を受けています．

参考文献

- [1] Y. T. Feng, D. R. J. Owen and D. Perić, A block conjugate gradient method applied to linear systems with multiple right-hand sides, *Comput. Methods Appl. Mech. Eng.*, **127** (1995), 203–215.
- [2] K. Abe and S.-L. Zhang, A variable preconditioning using the SOR method for GCR-like methods, *Int. J. Numer. Anal. Model.*, **2** (2005), 147–161.
- [3] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners*, Springer, New York, 2008.

行列方程式に対する低ランク型 Krylov 部分空間法のための前処理について

佐竹 祐樹¹, 曾我部 知広², 剣持 智哉², 張 紹良²

¹ 北海道大学情報基盤センター, ² 名古屋大学大学院工学研究科

e-mail: satake@iic.hokudai.ac.jp

1 はじめに

線形行列方程式は、信号処理や制御理論をはじめとした幅広い分野に現れる。科学技術計算において、特に大規模な問題に対する数値解法の高速化が重要な課題である。

一般に、行列方程式の係数行列が疎行列であっても、係数行列と同等のサイズの解は密行列となる。したがって、大規模な行列方程式に対して Krylov 部分空間法を適用する場合、大規模な密行列の計算が必要となり、多くの記憶容量を要する。低ランク型 Krylov 部分空間法は、反復毎に現れる密行列を低ランク行列で近似することで省メモリに計算可能な数値解法である（例えば [1]）。

低ランク型 Krylov 部分空間法は、省メモリに計算できる一方で、低ランク近似によって収束性が悪化する恐れがある。したがって、前処理による収束性の改善が重要である。本研究では、線形行列方程式を連立 1 次方程式とみなした際の係数行列に現れる行列構造（テンソル構造）に着目し、この構造に基づいた前処理手法の導出を行う。また、導出したいくつかの前処理を数値実験で比較する。

2 低ランク型 Krylov 部分空間法

本稿では、線形行列方程式の一種である Sylvester 方程式

$$AX + XB = C \quad (1)$$

を考える。ここで、 $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, $C \in \mathbb{R}^{n \times m}$ は既知行列、 $X \in \mathbb{R}^{n \times m}$ は未知行列である。また、右辺行列は低ランク、つまり、 $C = C_1 C_2^\top$, $C_1 \in \mathbb{R}^{n \times r}$, $C_2 \in \mathbb{R}^{m \times r}$, $r \ll m, n$ とする。

Sylvester 方程式 (1) に対する前処理付き低ランク BiCGSTAB 法のアルゴリズムを Algorithm 1 に示す。アルゴリズム内の計算に関する詳細な説明は紙面の都合上省略する（詳しくは [1] を参照されたい）が、反復毎に更新する各行列を低ランク行列として保持することで省メモリに計算できる。反復回数の増加とともに各行列のランクが増加するが、低ランク行列をさらに低ランクな行列で近似する打ち切り作用素 \mathcal{T} を適用することでランクを小さくし、所要メモリの増加を防ぐ。また、 \mathcal{M}^{-1} は前処理の計算を表しており、本研究では、前処理の計算方法について議論する。

3 テンソル構造に基づいた前処理

Sylvester 方程式 (1) は、ベクトル化によって連立 1 次方程式

$$\mathcal{F}x = c, \quad \mathcal{F} := I_m \otimes A + B^\top \otimes I_n \in \mathbb{R}^{mn \times mn} \quad (2)$$

に書き換えられる。ここで、 \otimes はテンソル積 (Kronecker 積) を表し、 I_n は n 次単位行列、 $x, c \in \mathbb{R}^{mn}$ はそれぞれ X, C のベクトル化を表す。係数行列 \mathcal{F} はテンソル積で表される行列の和になっており、本稿ではこの構造をテンソル構造と呼ぶ。

連立 1 次方程式に対する一般的な前処理では、係数行列を単位行列に近づけるような変換を行う前処理行列を用いることで収束性の改善を行う。一方で行列方程式の場合、これに加えて、所要メモリ

Algorithm 1 Sylvester 方程式に対する前処理付き低ランク BiCGSTAB 法

Input: $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$. Right-hand side $C \in \mathbb{R}^{n \times m}$ in low-rank format, $\tilde{R} \in \mathbb{R}^{n \times m}$ in low-rank format (e.g., $\tilde{R} = C$). Truncation operator \mathcal{T} .

Output: Approximate solution $X \in \mathbb{R}^{n \times m}$.

```
1:  $X^{(0)} = 0, R^{(0)} = C, P^{(0)} = R^{(0)}, \hat{P}^{(0)} = \mathcal{M}^{-1}(P^{(0)}), Q^{(0)} = A\hat{P}^{(0)} + \hat{P}^{(0)}B, \rho^{(0)} = \langle \tilde{R}, R^{(0)} \rangle$ 
2: for  $k = 0, 1, 2, \dots$  until convergence do
3:    $\alpha^{(k)} = \rho^{(k)} / \langle \tilde{R}, Q^{(k)} \rangle$ 
4:    $S^{(k)} = R^{(k)} - \alpha^{(k)} Q^{(k)},$  Optionally:  $S^{(k)} \leftarrow \mathcal{T}(S^{(k)})$ 
5:    $\hat{S}^{(k)} = \mathcal{M}^{-1}(S^{(k)}),$  Optionally:  $\hat{S}^{(k)} \leftarrow \mathcal{T}(\hat{S}^{(k)})$ 
6:    $T^{(k)} = A\hat{S}^{(k)} + \hat{S}^{(k)}B,$  Optionally:  $T^{(k)} \leftarrow \mathcal{T}(T^{(k)})$ 
7:    $\omega^{(k)} = \langle T^{(k)}, S^{(k)} \rangle / \langle T^{(k)}, T^{(k)} \rangle$ 
8:    $X^{(k+1)} = X^{(k)} + \alpha^{(k)} \hat{P}^{(k)} + \omega^{(k)} \hat{S}^{(k)},$   $X^{(k+1)} \leftarrow \mathcal{T}(X^{(k+1)})$ 
9:    $R^{(k+1)} = C - AX^{(k+1)} - X^{(k+1)}B,$  Optionally:  $R^{(k+1)} \leftarrow \mathcal{T}(R^{(k+1)})$ 
10:  Check convergence
11:   $\rho^{(k+1)} = \langle \tilde{R}, R^{(k+1)} \rangle$ 
12:   $\beta^{(k)} = \rho^{(k+1)} / \rho^{(k)} \cdot \alpha^{(k)} / \omega^{(k)}$ 
13:   $P^{(k+1)} = R^{(k+1)} + \beta^{(k)} P^{(k)},$   $P^{(k+1)} \leftarrow \mathcal{T}(P^{(k+1)})$ 
14:   $\hat{P}^{(k+1)} = \mathcal{M}^{-1}(P^{(k+1)}),$  Optionally:  $\hat{P}^{(k+1)} \leftarrow \mathcal{T}(\hat{P}^{(k+1)})$ 
15:   $Q^{(k+1)} = A\hat{P}^{(k+1)} + \hat{P}^{(k+1)}B,$  Optionally:  $Q^{(k+1)} \leftarrow \mathcal{T}(Q^{(k+1)})$ 
16: end for
```

の観点から、変換後の係数行列がテンソル構造をもつように前処理行列を構築する必要がある。そこで本研究では、前処理行列を

$$\mathcal{M} = M \otimes N \quad (3)$$

とおくことで、変換後の係数行列 $\tilde{\mathcal{F}} := \mathcal{F}\mathcal{M}^{-1}$ にテンソル構造を与えることを考える。また、(3) の構造により、Algorithm 1 の 1, 5, 14 行目でランクを増加させることなく前処理の計算ができる。

前処理行列が (3) の構造となる既存手法として、NKP (Nearest Kronecker Product) 前処理 [2] が知られている。NKP 前処理では、 $N = \nu_0 I_n + \nu_1 A$, $M = \mu_0 I_m + \mu_1 B^\top$ とおき、 $\|\mathcal{F} - M \otimes N\|_F$ を最小にするパラメータ $\nu_0, \nu_1, \mu_0, \mu_1$ を求める問題に帰着する。本研究では、これらのパラメータと \mathcal{M}^{-1} の計算 (M^{-1}, N^{-1} の計算) 方法を変化させて数値実験を行い、それらの結果を比較した。

4 まとめ

大規模行列方程式に対する低ランク型 Krylov 部分空間法のための前処理手法を導出した。アルゴリズムの詳細と数値実験の結果については、講演時に報告する。

謝辞 本研究は JSPS 科研費 JP23K19951 の助成を受けたものです。

参考文献

- [1] D. Kressner and C. Tobler, SIMAX, 32(4)(2011), pp. 1288–1316.
- [2] A. N. Langville and W. J. Stewart, NLAA, 11(8-9)(2004), pp. 723–752.