

Symplectic Gram-Schmidt 法の J -直交性

慶應義塾大学 理工学研究科 松尾 洋一

慶應義塾大学 理工学部 野寺 隆

Yoichi Matsuo

Graduate School of Science and Technology, Keio University

Takashi Nodera

Faculty of Science and Technology, Keio University

1 背景

特殊な構造を持つ行列の固有値を求めることはよくある。例えば、最適制御理論から導出される Riccati 方程式の解は、Hamiltonian 行列の固有値・固有ベクトルを求めることで数値的に解くことが出来る [5]。このように、Hamiltonian 行列の固有値問題は非常に重要であり、現在までにいくつかの研究が行われている [1, 3, 6]。これまでに、Lanczos 法 [10, 15], Arnoldi 法 [2], Jacobi-Davidson 法 [7, 13, 16] などの、様々な固有値問題の解法が提案されてきた。その中で、Symplectic 法という手法が提案された。

Symplectic 法は、行列 X の重要な構造を保持したまま計算することが可能である。Van Loan [6] によれば、Hamiltonian 行列に対しては、QR 法と比べて、SR 法は約 $1/4$ の計算コストと記憶領域だけを必要とすることが報告されている。また大規模疎行列の固有値問題に対しては、Symplectic Lanczos 法が提案されている [3]。この手法では、Symplectic なベクトル列を生成するために、Symplectic Gram-Schmidt (SGS) 法 SGS 法が使われている。SGS 法は、与えられた行列 X に対して $X = SR$ を満たす Symplectic 行列 S と、上三角行列 R を計算する手法であり、SR 分解を行う。

これまで複数の SR 分解のアルゴリズムが提案されており、Symplectic Householder 法と Symplectic Givens 回転を用いた SR 手法がある [6]。この手法は、Householder 法を用いた QR 法と似ている。Salam [12] は、この手法が Modified SGS による SR 分解と数学的に同値であることを証明した。またその他に、Classical SGS (CSGS) 法や Modified SGS (MSGs) 法を用いた SR 分解がある [11]。

このように、SGS 法も応用範囲の広い手法であるが、CGS 法を用いた QR 分解と比べ、SGS 法を用いた SR 分解に関する研究は少ない。Salam [11] によると、SGS 法はパラメータの取り方により J -直交精度が大きく変化することが指摘されているが、手法自体の誤差解析などは報告されていない。

そこで、本稿では J -直交行列の計算誤差を解析するとともに、再直交化の条件に関する検討を行った。そして、CSGS 法を高速化するために、ブロック化した Block Symplectic Gram-Schmidt (BSGS) 法を提案する。最後に、数値実験を用いて提案手法の有効性を示す。

Algorithm 1 Elementary SR Factorization

Require: $A_1 = [\mathbf{a}_1, \mathbf{a}_2]$
Ensure: $S_1 = [\mathbf{s}_1, \mathbf{s}_2], R_1 = [r_{11}, r_{12}, r_{21}, r_{22}]$

 Choose $r_{11} \in \mathbb{R}, \mathbf{s}_1 = \mathbf{a}_1 / r_{11}$

 Choose $r_{12} \in \mathbb{R}, \mathbf{y} = \mathbf{a}_2 - r_{12} \mathbf{s}_1$
 $r_{22} = \mathbf{s}_1^T J \mathbf{y}$
 $\mathbf{s}_2 = \mathbf{y} / r_{22}$

1.1 表記法

本節では、SGS 法に関わる手法についての表記について述べる。行列サイズが $n \times n$ の単位行列 I_n と零行列 0_n に対して、行列 $J \in \mathbb{R}^{2n \times 2n}$ を次式で定義する。

$$J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix} \quad (1)$$

ただし、 $J^T = J^{-1} = -J$ である。そして、2つのベクトル $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{2n}$ に対して、 J -内積を次のように定める。

$$\langle \mathbf{x}, \mathbf{y} \rangle_J = \mathbf{x}^T J \mathbf{y} \quad (2)$$

また、行列 M に対して、 M^J を次のように定義する。

$$M^J = J^T M^T J \quad (3)$$

行列 S が次式を満たすとき、 S は Symplectic、あるいは J -直交であるという。

$$S^J S = J^T S^T J S = I \quad (4)$$

2 Symplectic Gram-Schmidt 法

2.1 算法

本節では、与えられたベクトル列から J -直交ベクトル列を生成する方法について述べる。まず、与えられた2つのベクトル列 $X_1 = [\mathbf{x}_1, \mathbf{x}_2], \mathbf{x}_i \in \mathbb{R}^n, i = 1, 2$ を J -直交させる Elementary SR Factorization (ESR) 法を Algorithm 1 に示す。ESR 法は与えられた行列 X_1 に対して、次のように計算することになる。

$$\begin{aligned} \mathbf{s}_1 &= \frac{\mathbf{x}_1}{r_{11}} \\ \mathbf{y} &= \mathbf{x}_2 - r_{12} \mathbf{s}_1 \\ r_{22} &= \mathbf{s}_1^T J \mathbf{y} \\ \mathbf{s}_2 &= \frac{\mathbf{y}}{r_{22}} \end{aligned} \quad (5)$$

ただし、 r_{11}, r_{12} は任意の実数である。以上をまとめると次式が成り立つ。

$$X_1 = S_1 R_1 \quad (6)$$

ただし、行列 $S_1 = [\mathbf{s}_1, \mathbf{s}_2]$ は J -直交行列であり、 $R_1 = [r_{11}, r_{12}, r_{21}, r_{22}]$ は上三角行列である。ここで、 r_{11}, r_{12} は任意の実数である。Salam [11] は、このパラメータの選び方についての提案を行っている。

- ESR1 法: $r_{11} = \|\mathbf{x}_1\|, r_{12} = 0$
- ESR2 法: $r_{11} = \|\mathbf{x}_1\|, r_{12} = \mathbf{s}_1^T \mathbf{x}_2$

- ESR3 法: $r_{11} = \|\mathbf{x}_1^T J \mathbf{x}_2\|, r_{12} = 0$

特に ESR2 法において, $r_{11} = \|\mathbf{x}_1\|, r_{12} = \mathbf{s}_1^T \mathbf{x}_2$ であるので, 式 (5) より次式が成り立つ.

$$\mathbf{s}_1 = \frac{\mathbf{x}_1}{r_{11}} = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} \quad (7)$$

$$\mathbf{s}_2 = \frac{\mathbf{y}}{r_{22}} = \frac{\mathbf{x}_2 - \mathbf{s}_1^T \mathbf{x}_2 \mathbf{s}_1}{\mathbf{s}_1^T J \mathbf{x}_2 - \mathbf{s}_1^T \mathbf{s}_1^T \mathbf{x}_2 \mathbf{s}_1} = \frac{\mathbf{x}_2 - \mathbf{s}_1^T \mathbf{x}_2 \mathbf{s}_1}{\mathbf{s}_1^T J \mathbf{x}_2} \quad (8)$$

式 (7) と式 (8) より, 次式が成立する.

$$\mathbf{s}_1^T \mathbf{s}_2 = \frac{\mathbf{s}_1^T \mathbf{x}_2 - \mathbf{s}_1^T \mathbf{s}_1^T \mathbf{x}_2 \mathbf{s}_1}{\mathbf{s}_1^T J \mathbf{x}_2} = 0 \quad (9)$$

従って ESR2 法を用いると, $\mathbf{s}_1$ と $\mathbf{s}_2$ は $\mathbf{s}_1 \perp \mathbf{s}_2$ が成り立つ. これは $\|\mathbf{s}_2\|$ が最小になる選び方であるので, これらの中で最も数値的に安定していることが報告されている [11]. 本稿では, ESR2 法を使用することにする.

与えられた行列 $A \in \mathbb{R}^{2n \times 2n}$ に対して, 次のような分解をする手法の 1 つに Classical Symplectic Gram-Schmidt (CSGS) 法がある.

$$A = SR \quad (10)$$

ただし, S は J -直交行列であり, R は上三角行列である. 与えられた J -直交行列 S に対して, CSGS 法によるベクトル列 $X = [\mathbf{x}_1, \mathbf{x}_2]$ の J -直交化は次のようになる.

$$H_{12} = S^J X \quad (11)$$

$$Y = X - SH_{12} \quad (12)$$

$$Y \rightarrow SR \quad (\text{by ESR}) \quad (13)$$

式 (12) と式 (13) からわかるように, CSGS 法の計算は CGS 法に似ている. しかし, CGS 法と異なり正規化を行わず, 代わりに ESR 法を用いて $W_i = [\mathbf{w}_{2i-1}, \mathbf{w}_{2i}]$ の関係を定めている. 以下これを繰り返すことによって, 式 (10) を得る. Algorithm 2 に CSGS 法の算法を示す.

CSGS 法において, 式 (12) と式 (13) が主となる部分であり, CGS 法の次式に対応している.

$$\mathbf{r} = Q^T \mathbf{x}$$

$$\hat{\mathbf{y}} = \mathbf{x} - Q\mathbf{r}$$

また, この算法によって生成されたベクトル列 $S_1, \dots, S_n, S_i = [\mathbf{s}_{2i-1}, \mathbf{s}_{2i}]$ は, 次式を満たす.

$$\mathbf{s}_{2i-1}^T J \mathbf{s}_{2i} = 1, \quad i = 1, \dots, n \quad (14)$$

$$\mathbf{s}_i^J \mathbf{s}_j = 0, \quad i \neq j, \quad \mathbf{s}_i^J \mathbf{s}_j = 1, \quad i = j \quad (15)$$

よって, 次式が成立する.

$$\dim \text{span}\{S_1, \dots, S_n\} = 2n \quad (16)$$

このように CSGS 法は与えられた行列と同様の空間を張るベクトル列を生成する. 次に, Modified Symplectic Gram-Schmidt (MSGS) 法について述べる. MSGS 法は, 与えられた

Algorithm 2 Classical Symplectic Gram-Schmidt Algorithm

Require: $X_1, \dots, X_n$
Ensure: $S = [S_1, \dots, S_n], R$
 $X_1 = S_1 R(1:2, 1:2)$
for $i = 2 : n$ **do**

 for $j = 1 : i - 1$ **do**

 $H_{i,j} = S_j^T X_i$

 end for
 $Y_i = X_i - \sum_{j=1}^{i-1} S_j H_{i,j}$
 $R(1:2(i-1), 2i-1:2i) = H_{j,i}$
if $(Y_i \neq 0)$ **then**

 $Y_i = S_i \dot{R}$ by ESR algorithm

else

exit

end if
 $R(2i-1:2i, 2i-1:2i) = \dot{R}$
end for

J -直交行列 S に対して、次式で J -直交化を行う。

$$Y = X \tag{17}$$

for $i = 1, \dots, j-1$

$$H_i = S_i^J Y \tag{18}$$

$$Y = Y - S_i H_i \tag{19}$$

end for

$$Y \rightarrow SR \quad (\text{by ESR}) \tag{20}$$

CSGS 法の場合と同様、MSGGS 法も MGS 法と似た手法である。また、正規化を行わずに ESR 法を用いて $Y_i = [\mathbf{y}_{2i-1}, \mathbf{y}_{2i}]$ の関係を定めている。Algorithm 3 に MSGGS 法の算法を示す。MGS 法の場合と同じように MSGGS 法においても、CSGS 法よりも MSGGS 法の方が、 J -直交行列の計算精度が高いことがわかっている [12]。

ここで、GS 法と SGS 法の違いについて述べる。

- Gram-Schmidt 法

- $X \in \mathbb{R}^{n \times n} \rightarrow QR$

$$\text{span}(X) = \text{span}(\mathbf{q}_1, \dots, \mathbf{q}_n), \text{ where } \langle \mathbf{q}_i, \mathbf{q}_j \rangle = 0, i \neq j$$

- Symplectic Gram-Schmidt 法

- $X \in \mathbb{R}^{2n \times 2n} \rightarrow SR$

$$\text{span}(X) = \text{span}(S_1, \dots, S_n)$$

$$\Leftrightarrow \text{span}(X) = \text{span}([\mathbf{s}_1, \mathbf{s}_2], [\mathbf{s}_3, \mathbf{s}_4], \dots, [\mathbf{s}_{2n-1}, \mathbf{s}_{2n}])$$

$$\langle S_i, S_j \rangle_J = 0, i \neq j$$

Algorithm 3 Modified Symplectic Gram-Schmidt Algorithm

Require: $X_1, \dots, X_n$ **Ensure:** $S_1, \dots, S_n, R$ $X_1 = S_1 R(1:2, 1:2)$ **for** $i = 2 : n$ **do** $Y_i = X_i$ **for** $j = 1 : i - 1$ **do** $H_{i,j} = S_j^J Y_i$ $Y_i = Y_i - \sum_{j=1}^{i-1} S_j H_{i,j}$ **end for** $R(1:2(i-1), 2i-1:2i) = H_{j,i}$ **if** $(Y_i \neq 0)$ **then** $Y_i = S_i \hat{R}$ by ESR algorithm**else****exit****end if** $R(2i-1:2i, 2i-1:2i) = \hat{R}$ **end for**

GS 法は与えられた行列 $X \in \mathbb{R}^{n \times n}$ に対して、 n 本の正規直交基底列を生成する算法である。このとき行列 X が張る空間と正規直交基底列が張る空間は同じになる。一方、SGS 法は、与えられた行列 $X \in \mathbb{R}^{2n \times 2n}$ に対して、 n 個の J -直交行列 S_i を生成する。このとき行列 X が張る空間と J -直交行列 S_i が張る空間は同じになる。また、ESR2 法を用いると、 $S_i = [s_{2i-1}, s_{2i}]$ において s_{2i-1} と s_{2i} は直交している。

2.2 J -直交性の崩れ

本節では、SGS 法による SR 分解の J -直交行列の数値的性質について述べる。一般的に QR 分解によって得られた直交行列 Q の計算精度は、次式のようにになる。ただし、行列 I は単位行列である。

$$\|I - Q^T Q\|_2 \quad (21)$$

Björck [4] によれば、丸め誤差による影響は QR 分解された行列 A の条件数に比例する。また、Stewart [14] によって再直交化するための指標が導入された。本節では、SGS 法に対して、 J -直交化の精度について考える。

まず、 J -直交行列の計算精度を次式で表すことにする。

$$\|I - S^J S\|_2 \quad (22)$$

Salam [11] は、次のような事柄の指摘をしている。ESR 法において r_{11} と r_{12} の選択の仕方により、 J -直交性が大きく変化して崩れることがあり、また、行列 X の条件数が 1 のように小さかったとしても、SGS 法においては J -直交性が崩れることもある。この詳細な理由は述べられていないが、一般的な条件数の定義を用いる代わりに $\text{cond}(X) = \sqrt{(X^T J X)}$ を用いることを提案している。

2.3 Symplectic Gram-Schmidt 法の再直交化

GS 法の再直交化も重要な研究の 1 つである [4, 14]. GS 法は直交化されるベクトル列が互いに平行に近い場合や丸め誤差の影響などにより、直交行列を正確に求められない場合がある. このような場合には、再直交化を行う. しかし、再直交化することで計算コストが増加するので、良い再直交化の条件を検討することが非常に重要になる. GS 法のステップによって直交化されたベクトル $\hat{\mathbf{y}}$ が、次式を満たすものとする.

$$\|\hat{\mathbf{y}}\| \geq \frac{1}{2}\|\mathbf{x}\| \quad (23)$$

このことより、 $\hat{\mathbf{y}}$ の直交化されていない成分の大きさが 2ε 以下となる. 逆に、この条件が満たされない場合は、再直交化すれば良いことがわかる [14].

しかし、SGS 法に対しては、これがうまく機能しないことが予想される. 2.2 節で示したように、 S のノルムが大きくなる傾向があるので、式 (23) を満たしているが、 J -直交性が崩れてしまうことが予測される. そのような場合においても再直交化が行われるように、次のような新しい再直交化条件を考える.

$$\|\mathbf{y}\| \leq \frac{1}{2}\|\mathbf{x}\| \quad (24)$$

この式を用いると、 $\mathbf{y}$ のノルムを上からの有界性が得られ、 J -直交化されたベクトルのノルムが大きくなることを防ぎ、 J -直交性が崩れることを回避することが出来る.

3 Block Symplectic Gram-Schmidt 法

本節では、CSGS 法のブロック化を提案する. Stewart [14] や Matsuo ら [8, 9] は、GS 法をブロック化することにより、正規直交基底列を高速に計算することが出来ることを示した. そこで、CSGS 法を高速化するのブロック化を考察する. まず、CSGS 法は式 (12) と式 (13) である. これらの式中の X を $X_{\text{block}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ で置き換えると、式 (12) と式 (13) は次式になる.

$$H_{12} = S^J X_{\text{block}} \quad (25)$$

$$Y = X_{\text{block}} - S H_{12}. \quad (26)$$

式 (25) と式 (26) により、 X_{block} は J -直交行列 S を用いて J -直交化することが出来る. しかし、これだけでは X_{block} 中のベクトル列どうしは J -直交化できていない. そこで X_{block} を、次式のように CSGS 法によって、 J -直交化させる.

$$X_{\text{block}} \rightarrow S R \quad (\text{by CSGS}) \quad (27)$$

そうすることで、CSGS 法をブロック化することが可能となる. Block Symplectic Gram-Schmidt (BSGS) 法の算法を Algorithm 4 に示す. CSGS 法のブロック化によって、BLAS を用いて高速に演算することが可能になる.

次に、CSGS 法、MSGs 法、BSGS 法の性質を比較する.

- Classical Symplectic Gram-Schmidt 法
 - 計算精度：低い
 - 並列性：高い
- Modified Symplectic Gram-Schmidt 法
 - 計算精度：高い

Algorithm 4 Block Symplectic Gram-Schmidt Algorithm

Require: $X = [X_{\text{block}_1}, \dots, X_{\text{block}_n}]$
Ensure: $S = [S_1, \dots, S_n], R$
 $X_{\text{block}_1} = S_1 R(1:2, 1:2)$
for $i = 2 : n$ **do**

 for $j = 1 : i - 1$ **do**

 $H_{i,j} = S_j^J X_{\text{block}_i}$

 end for

 $Y_i = X_{\text{block}_i} - \sum_{j=1}^{j=i-1} S_j H_{i,j}$

 $R(1:2(i-1), 2i-1:2i) = H_{j,i}$

 $Y_i = S_i \hat{R}$ by CSGS method

 $R(2i-1:2i, 2i-1:2i) = \hat{R}$
end for

– 並列性：低い

• Block Symplectic Gram-Schmidt 法

– 計算精度：低い

– 並列性：高い

GS 法の場合と同様に，MSGGS 法は CSGS 法より J -直交行列の計算精度が高い．BSGS 法は CSGS 法をブロック化しているため，CSGS 法と類似の性質を持っている．しかし，BSGS 法の X_{block_i} の列ベクトルは，式 (25) と式 (26) によって，まず行列 S に対して J -直交化を計算した後，式 (27) によって， X_{block_i} 中のベクトル列どうしを J -直交化しているため，一部 MSGGS 法を取り入れた形になっていると考えられる．よって，わずかながら，CSGS 法より計算精度が高くなっている可能性がある．

4 数値実験

本節では， J -直交性の崩れ，CSGS のブロック化の有効性を数値実験を用いて示す．以下の実験環境において数値実験を行った．

- OS : Ubuntu14.04 LTS
- CPU : Intel(R) Xeon(R) CPU E3-1270 V2 3.50GHz
- Memory : 16GB
- 精度 : 倍精度

数値実験の結果を示す表に使われる記号はそれぞれ次を表す．

- CSGS : Classical Symplectic Gram-Schmidt 法
- MSGS : Modified Symplectic Gram-Schmidt 法

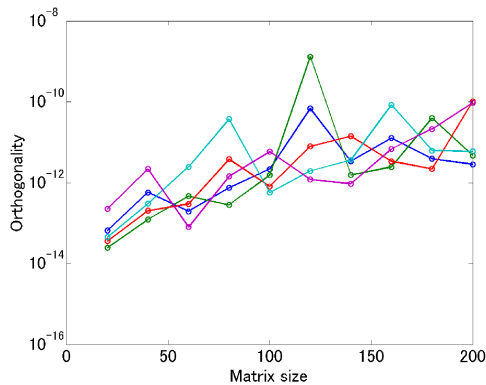


図 1 CGS 法による直交行列の計算誤差

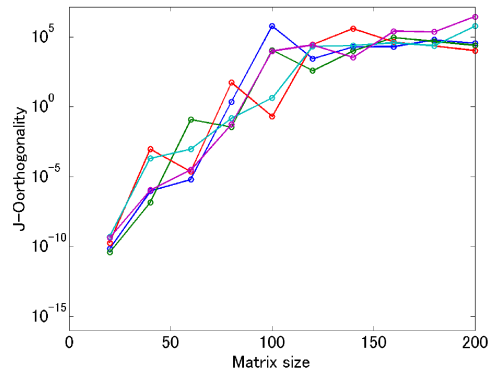


図 2 CS GS 法による J -直交行列の計算誤差

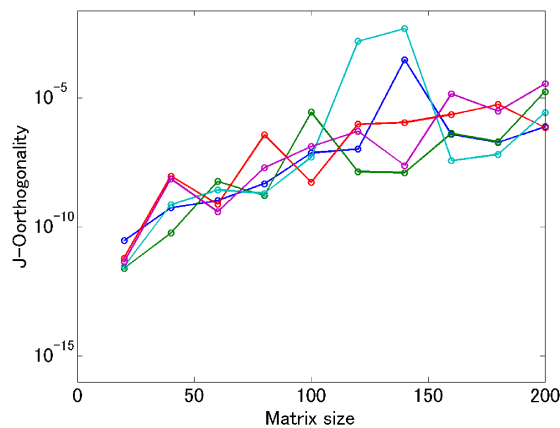


図 3 MSGS 法による J -直交行列の計算誤差

- BSGS : Block Symplectic Gram-Schmidt 法
- t : 計算時間 (秒)
- Accuracy : 式 (22) を用いて計算した J -直交行列の計算精度

4.1 数値実験 1

本節では, SGS 法の J -直交性の崩れと, 再直交化について, Matlab 上で数値実験を行った。テスト行列として, 区間 $[1:10]$ の一様分布に従うランダム値からなる Hamiltonian 行列を用いる。行列サイズは, それぞれ $20 \times 20, 40 \times 40, \dots, 200 \times 200$ の 10 個である。CSGS 法と MSGS 法を用いて SR 分解を行い, 式 (22) を用いて J -直交行列の計算精度をそれぞれを 5 回ずつ測った。計算結果を図 2, 図 3 に示す。図 2 において, 横軸は行列サイズ, 縦軸は式 (22) を用いて計算した J -直交行列の計算誤差の値である。それぞれの折れ線は, 各 1 回目から 5 回目の試行に対応している。比較のために, CGS 法を用いて同様の行列に対して QR 分解を行い, 式 (21) を用いて直交行列の計算精度を調べた結果を図 1 に示す。

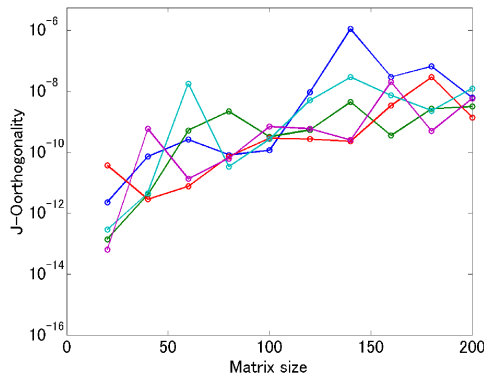


図 4 CSGS 法による J -直交行列の計算誤差の様子

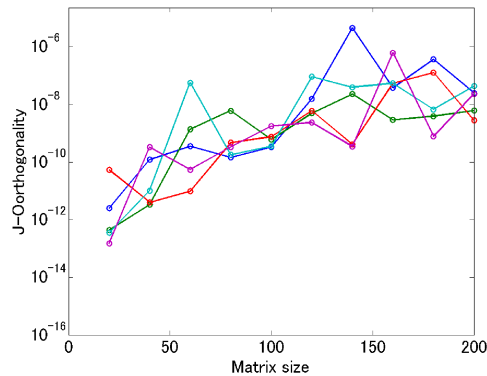


図 5 MSGS 法による J -直交行列の計算誤差の様子

図 1 からわかるように、10 から 200 程度のサイズ行列に対しての CGS 法は、 10^{-15} 程度の非常に高い精度で直交行列が計算できている。一方、図 2 からわかるように、SGS 法による J -直交行列の計算は、行列サイズが小さいものに対しては、GS 法と同程度の精度で計算できているが、行列サイズが 100 程度まで行くと、 J -直交行列が正確に計算できていないことがわかる。また、図 3 より、MSGS 法においても GS 法と同程度の直交精度は計算できていないことがわかる。しかしながら、CSGS 法よりも良く J -直交行列が計算できている。このことから MSGS 法が、CSGS 法よりも数値的に安定であることがわかる。そこでこれまでの条件に加えて、2.2 節で提案した式 (24) を用いて再直交化をする CSGS 法と MSGS 法を用いて SR 分解を行った。結果を図 4 と図 5 に示す。

図 4 より、再直交化付きの CSGS 法は図 2 と比べて、精度が高くなっていることがわかる。一方で、図 5 より、再直交化付きの MSGS 法は図 3 と比べてそれほど精度は高くない。また、どちらの手法も GS 法と比べて、未だに同程度の精度は得られていないことがわかる。これらの結果より、CSGS 法は著しく J -直交精度が低くなりやすいので、式 (24) は有用であるが、より厳しい条件が必要であるといえる。

4.2 数値実験 2

本節では、CSGS 法のブロック化の有効性を数値実験を C 言語を用いて示す。テスト行列として、行列サイズが 200×200 と 1000×1000 である、区間 $[1 : 10]$ の一様分布に従うランダム値からなる Hamiltonian 行列をテスト行列として用いる。CSGS 法、MSGS 法、BSGS 法を用いてテスト行列を SR 分解し、計算時間と式 (22) を用いて J -直交精度を計算した。ただし、全ての手法において再直交化を一度行った。数値実験の結果を表 1 と表 2 に示す。

表 1 より、BSGS 法は他の 2 つの手法と比べて、SR 分解を約 2 倍程度高速に計算することが出来ることがわかる。これはブロック化することにより、 J -直交行列を計算に使用する回数が減り、BLAS を用いることで高速に計算が出来るためであると考えられる。また、3 節で述べたように、CSGS 法と比べ、わずかではあるが J -直交行列の計算精度が向上している。しかしながら表 1 では、MSGS 法に対しても BSGS 法の J -直交行列の計算精度が高くなっている。

表 1 CSGS 法, MSGS 法, BSGS 法の比較 : Hamiltonian 行列 1

Method	m	t_m	Accuracy
CSGS	1	0.104	8.70e-06
MSGS	1	0.107	4.65e-06
BSGS	5	0.039	2.80e-06

表 2 CSGS 法, MSGS 法, BSGS 法の比較 : Hamiltonian 行列 2

Method	m	t_m	Accuracy
CSGS	1	21.24	1.55e-04
MSGS	1	21.40	1.03e-04
BSGS	20	2.50	3.74e-05
BSGS	50	3.21	7.14e-05

表 2 より, 同様に BSGS 法は他の 2 つの手法と比べて, 約 10 倍程度高速になっている. 特に表 1 と比べて, CSGS 法との速度比が約 5 倍程度大きくなっている. これは扱う行列サイズが大きくなり, BLAS によるパフォーマンスの向上が大きくなったためである. また, J -直交性の計算精度については, 表 1 と比べて, より大きい幅で精度が向上していることがわかる. これは, 4.1 節からわかるように, 行列サイズが大きい問題の方が, J -直交行列の計算が不安定であるためと考えられる.

5 結論

本稿では, SGS 法と, J -直交性の崩れについて述べたあと, 再直交化条件の検討を行った. そして, CSGS 法のブロック化を提案した.

GS 法と異なり, SGS 法は行列 J の影響によって, 直交性が崩れる可能性があることを述べた. そして, 数値実験を用いて J -直交性が崩れやすいことを確認した. 特に CSGS 法では, 200 次元程度のサイズの行列の SR 分解でさえも, まったく正確に計算できていないことがわかった.

さらに CSGS 法においては, 直交化されたノルムが非常に大きくなることを確認した. これらのことより, CSGS 法においては, 再直交化が非常に重要であることがわかった. GS 法には, 再直交化の適用に関する判定条件が存在しているが, それをそのまま SGS 法に適用しても, 直交化されたベクトルノルムが大きくなるので, 有用でないことを示した. そこで, 本稿では新しい再直交化の条件を提案し, 数値実験を行った. 4.1 節の数値実験の結果より, CSGS 法, MSGS 法に対して, この再直交化条件が有用であることを示した.

また, CSGS 法を高速化するために, CSGS 法のブロック化を提案した. ブロック化することにより, CSGS 法と比べて, 計算順序が多少入れ替わる. これにより, わずかではあるが, BSGS 法の方が計算された J -直交行列の計算精度が向上する可能性があることを示した. そして, 4.2 節の数値実験により, ブロック化することで CSGS 法を高速化し, 計算精度も向上したことを検証し, 提案手法の有効性を示した.

参考文献

- [1] Ammar, G. and Benner, P.: On Hamiltonian and Symplectic Hessenberg Forms, *Linear Algebra Appl.*, Vol. 149, No. 1, pp. 55–72 (1991).
- [2] Arnoldi, W. E.: The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem, *Quarterly of Applied Mathematics*, Vol. 9, pp. 17–29 (1951).
- [3] Benner, P. and Fassbender, H.: An Implicitly Restarted Symplectic Lanczos Method for the Hamiltonian Eigenvalue Problem, *Linear Algebra Appl.*, Vol. 263, pp. 75–111 (1997).
- [4] Björck, A.: Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization, *BIT*, Vol. 7, pp. 1–21 (1967).
- [5] Bunse-Gerstner, A. and Mehrmann, V.: A Symplectic QR Like Algorithm for the Solution of the Real Algebraic Riccati Equation, *IEEE Trans. Autom. Control*, Vol. AC31, pp. 1104–1113 (1986).
- [6] Loan, C. V.: A Symplectic Method for Approximating All the Eigenvalues of a Hamiltonian Matrix, *Linear Algebra Appl.*, Vol. 61, pp. 223–251 (1984).
- [7] Matsuo, Y., Guo, H. and Arbenz, P.: Experiments on a Parallel Nonlinear Jacobi-Davidson Algorithm, *Procedia Computer Science*, Vol. 29, pp. 566–575 (2014).
- [8] Matsuo, Y. and Nodera, T.: The Optimal Block-Size for the Block Gram-Schmidt Orthogonalization, *J. Sci. Tech.*, Vol. 49, pp. 569–584 (2011).
- [9] Matsuo, Y. and Nodera, T.: An Efficient Implementation of the Block Gram-Schmidt Method, *Anziam J.*, Vol. 54, pp. C394–409 (2013).
- [10] Saad, Y.: *Iterative Methods for Sparse Linear Systems*, SIAM Second Edition (2003).
- [11] Salam, A.: On Theoretical and Numerical Aspects of Symplectic Gram-Schmidt-like Algorithms, *Numer. Algo.*, Vol. 39, pp. 437–462 (2005).
- [12] Salam, A. and Al-Aidarous, E.: Equivalence Between Modified Symplectic Gram-Schmidt and Householder SR Algorithms, *BIT Numer. Math.*, Vol. 54, pp. 283–302 (2014).
- [13] Sleijpen, G. L. G. and van der Vorst, H. A.: A Jacobi–Davidson iteration method for linear eigenvalue problems, *SIAM Rev.*, Vol. 40, pp. 267–293 (2000).
- [14] Stewart, G. W.: Block Gram-Schmidt Orthogonalization, *SIAM J. Sci. Comput.*, Vol. 31, pp. 761–775 (2008).
- [15] Teramoto, K. and Nodera, T.: A Note on Lanczos Algorithm for Computing PageRank, *CMCGS 2014*, pp. 12–15 (2014).
- [16] Voss, H.: A Jacobi–Davidson Method for Nonlinear and Nonsymmetric Eigenproblems, *Comput. Struct.*, Vol. 85, No. 17–18, pp. 1284–1292 (2007).