Block symplectic Gram–Schmidt method

Yoichi Matsuo[1] Takashi Nodera[2]

(Received 5 March 2015; revised 11 February 2016)

Abstract

For large scale linear problems, it is common to use the symplectic Lanczos method which uses the symplectic Gram–Schmidt method to compute symplectic vectors. However, previous studies showed that the selection process of the parameter in the symplectic Gram–Schmidt method is flawed, as it results in a partially destroyed J-orthogonality of the J-orthogonal matrix. We explore a block type symplectic Gram–Schmidt method and a new condition for the reorthogonalization to maintain J-orthogonality and to more accurately compute symplectic factorization. Applying the block size scheme to this method, we develop a new procedure for computing symplectic vectors.

http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/9380 gives this article, Published February 22, 2016, as part of the Proceedings of the 17th Biennial Computational Techniques and Applications Conference. ISSN 1446-8735.

Contents

1 Introduction C417

2 Symplectic Gram–Schmidt method C418

3 Block symplectic Gram–Schmidt method C421
3.1 Re-orthogonalization C421
3.2 Optimal block size C423

4 Numerical experiments C423
4.1 Experiment one C425
4.2 Experiment two C425

5 Conclusion C428

References C429

1 Introduction

The orthogonalization process of QR factorization by the Gram–Schmidt (GS) method is arguably one of the most important processes in linear algebraic computation and there are numerous studies on this subject [4, 7, 8, 11, 12].

Orthogonalization with the GS method is also used in symplectic methods which are structure-preserving and are used to solve eigenvalue problems arising from special matrices like the Hamiltonian matrix. In scientific computation, the eigenvalue problem of the Hamiltonian matrix is an important and well-studied topic [1, 2, 6]. One applications of the symplectic method is solving the Ricatti equation arising from control theory [3, 5]. The symplectic method enables us to compute eigenvalues quickly by using the structure of a matrix, unlike the Householder QR or Lanczos methods. For a given coefficient matrix A, the symplectic Gram–Schmidt (SGS) method computes the symplectic matrix S and triangular matrix R which satisfy $A = SR$. Since this method preserves the important structure of A, it enables us to comupute eigenvalues more rapidly. According to Van Loan [6], for a Hamiltonian matrix, the SGS requires only about 25% of the floating point operations of the Householder QR method.

The SR procedure is very similar to the Householder QR algorithm. Salam [10] proved that SR is equivalent to the modified symplectic Gram–Schmidt (MSGS) method. Another method is SR factorizaton by the classical symplectic Gram–Schmidt (CSGS) method [9]. However, there are fewer numerical experiments documented on the SGS compared to the GS decomposition for the QR, Arnoldi and Lanczos methods. In Section 2, we summarize the SGS method.

In this article we explore the possibility of using the block symplectic Gram–Schmidt (BSGS) method by blocking the CSGS method. Section 3 proposes the BSGS method. The block Gram–Schmidt (BGS) algorithm is a standard generalization of the classical Gram–Schmidt algorithm. Stewart [12] and Matsuo et al. [7] showed how the computation time of the QR factorization is shortened by employing the BGS method. By blocking the CSGS method, the BSGS method should enable a more rapid SR factorization. However, since the optimal block size m is not consistent when employing the BGS method, it is necessary to determine m. When the BGS method is used, m must be determined accurately through trial and error. Section 4 and Section 5 discuss numerical experiments which evaluate the effectiveness of our proposed algorithm.

2 Symplectic Gram–Schmidt method

The first step is to define the matrix

$$J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \tag{1}$$

where $J^T = J^{-1} = -J$. Then, the J-product is defined for the vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{2n}$ by

$$\langle \mathbf{x}, \mathbf{y} \rangle_J = \mathbf{x}^T J \mathbf{y} \,. \tag{2}$$

Let

$$M^J = J^T M^T J \,, \tag{3}$$

and let matrix S be symplectic or J-orthogonal when

$$S^J S = J^T S^T J S = I \,. \tag{4}$$

The elementary symplectic factorization (ESR) which J-orthogonalizes $X_1 = [\mathbf{x}_1, \mathbf{x}_2]$ into the J-orthogonal matrix $S_1 = [\mathbf{s}_1, \mathbf{s}_2]$, for $\mathbf{x}_i \in \mathbb{R}^n$ with $i = 1, 2$, is

$$\begin{cases} \mathbf{s}_1 = \mathbf{x}_1 / r_{11} \,, \\ \mathbf{y} = \mathbf{x}_2 - r_{12} \mathbf{s}_1 \,, \\ r_{22} = \mathbf{s}_1^T J \mathbf{y} \,, \\ \mathbf{s}_2 = \mathbf{y} / r_{22} \,, \end{cases} \tag{5}$$

where r_{11} and r_{12} are arbitrary real values. There are several way to choose r_{11} and r_{12}:

- ESR1, $r_{11} = \|\mathbf{x}_1\|$, $r_{12} = 0$,
- ESR2, $r_{11} = \|\mathbf{x}_1\|$, $r_{12} = \mathbf{s}_1^T \mathbf{x}_2$,
- ESR3, $r_{11} = \|\mathbf{x}_1^T J \mathbf{x}_2\|$, $r_{12} = 0$.

According to Salam [10], the choice of r_{11} and r_{12} is an influential factor in the accuracy of the J-orthogonality of the SR factorization and the ESR2 method is the most stable because s_1 and s_2 are orthogonal to each other.

From equation (5),

$$X_1 = S_1 R_1 \,, \quad R_1 = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} , \tag{6}$$

Algorithm 1: Elementary SR factorization.

Data: $X_1 = [\mathbf{x}_1, \mathbf{x}_2]$
Result: $S_1 = [\mathbf{s}_1, \mathbf{s}_2]\,,\ R_1 = [r_{11}, r_{12}, 0, r_{22}]$

```
begin
    Choose r_11 ∈ ℝ , s_1 = x_1/r_11 ;
    Choose r_12 ∈ ℝ , y = x_2 − r_12 s_1 ;
    r_22 = s_1^T J y ;
    s_2 = y/r_22 ;
end
```

where S_1 is the J-orthogonal matrix and R_1 is an upper triangular matrix. Algorithm 1 describes the ESR method.

The CSGS algorithm is very similar to the CGS algorithm. For an $2n \times 2n$ matrix $X = [X_1, X_2, \dots, X_n]$, the CSGS method factorizes X into the J-orthogonal matrix S and upper triangular matrix R through

$$H = S^J X_i\,, \tag{7}$$

$$Y_i = X_i - SH\,, \tag{8}$$

$$Y_i \rightarrow S_i R_i\,, \quad \text{(by ESR)} \tag{9}$$

$$S = [S, S_i]\,, \quad R \leftarrow R, R_i, H\,. \tag{10}$$

From equations (7) and (10) it is seen that the CSGS method is very similar to the CGS method. However, in the CSGS method the vectors Y are normalized by the ESR instead of by the norm of Y vectors. Repeating equations (7)–(10) for $i = 1, \dots, n$ results in

$$X = SR\,. \tag{11}$$

The J-orthogonalized vectors $S_1, \dots, S_n$, with $S_i = [\mathbf{s}_{2i-1}, \mathbf{s}_{2i}]$, in the CSGS method satisfy

$$\mathbf{s}_{2i-1}^T J \mathbf{s}_{2i} = 1\,, \quad i = 1, \ldots, n\,, \tag{12}$$

$$S_i^J S_j = \begin{cases} 0, & i \neq j\,, \\ 1, & i = j\,. \end{cases} \tag{13}$$

3 Block symplectic Gram–Schmidt method

In this section, a block CSGS method is explored for speeding-up SR factorization. First, matrix X in equations (7)–(10) is replaced with $X_{\text{block}} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and

$$H = S^J X_{\text{block}_i}\,, \tag{14}$$

$$Y_i = X_{\text{block}_i} - SH\,. \tag{15}$$

Using equations (14) and (15), a matrix X_{block} is J-orthogonalized against the previous J-orthogonalized matrix S. However, these steps alone are not enough to create a J-orthogonalized matrix because the vectors in Y are not J-orthogonalized against each other. This makes it necessary to add one more step to create J-orthogonalization for vectors in Y:

$$Y_i \rightarrow S_i R_i\,, \quad \text{(by CSGS)} \tag{16}$$

$$S = [S, S_i]\,, \quad R \leftarrow R, R_i, H\,. \tag{17}$$

By relation (17), Y is J-orthogonalized. Algorithm 2 gives the BSGS algorithm.

3.1 Re-orthogonalization

According to Stewart [12], employing full re-orthogonalization is enough for maintaining the orthogonality of computed vectors when employing the GS

Algorithm 2: Block symplectic Gram–Schmidt algorithm.

Data: $X=[X_{\mathrm{block}_1},\ldots,X_{\mathrm{block}_n}]$
Result: $S=[S_1,\ldots,S_n]$, R
begin
  $X_{\mathrm{block}_1}=S_1R(1:2,1:2)$;
  for $i=2:n$ **do**
    for $j=1:i-1$ **do**
      $H_{i,j}=S_j^JX_{\mathrm{block}_i}$;
    end
    $Y_i=X_{\mathrm{block}_i}-\sum_{j=1}^{j=i-1}S_jH_{i,j}$;
    $R(1:2(i-1),2i-1:2i)=H_{j,i}$;
    $Y_i\rightarrow S_iR_i$ by CSGS method;
    $R(2i-1:2i,2i-1:2i)=R_i$;
  end
end

method. The condition for re-orthogonalization is

$$\|\hat{\mathbf{y}}\|\geqslant\frac{1}{2}\|\mathbf{x}\|\,. \tag{18}$$

If an orthogonalized vector $\hat{\mathbf{y}}$ does not satisfy this condition, then re-orthogonalization is employed. However, when running the SGS method, the condition may fail because the norms of J-orthogonalized vectors tend to increase as the SGS steps proceed. This is because the SGS method is unable to normalize computed vector $\hat{\mathbf{y}}$. To address this issue the ESR method must be utilized instead. The s_{2i-1} vector satisfies

$$\|\mathbf{s}_{2i-1}\|\geqslant 1\,. \tag{19}$$

Even though the computed vector $\hat{\mathbf{y}}$ lacks J-orthogonality, $\hat{\mathbf{y}}$ satisfies condition (18). This makes it possible to propose a new condition for re-orthogonalization instead of using condition (18):

$$\|\hat{y}\| \leqslant \frac{1}{2}\|x\|\,. \tag{20}$$

Through this condition, the norm of the computed vectors is controlled, and re-orthogonalization is employed.

3.2 Optimal block size

Through a blocking procedure, it is possible to compute SR factorization quickly. However, the computation time is dependent on block size m. Moreover, since the optimal block size m is not consistent when employing the BSGS, it is mandatory to determine m. There is no unique m for any matrix X when the BSGS is used, and it is necessary to determine m accurately, through trial and error.

The next step is to determine the optimal block size. Algorithm 3 gives the method used to estimate optimal block size. The total BSGS computation time with block size m is approximated from a polynomial function using sample points $b[i]$ which are estimated by observing the sample computation time a. We determine that the m which minimizes the approximated BSGS computation time is the optimal block size. Matsuo et al. [8] provided more details for determining the block size.

4 Numerical experiments

In this section the BSGS with Algorithm 2 and Algorithm 3, CSGS and MSGS are tested [9]. The numerical environment is

- Intel(R) Xeon(R) CPU E3-1270 V2 3.50 GHz;
- 16 GB memory.

Algorithm 3: The new method for estimating optimal block size.

Data: $X \in \mathbb{R}^{n\times n}$
Result: m

begin
  for $i = 1:5$ **do**
    $m_i := 2^{i-1}$;
    for $j = 0:1$ **do**
      start:=get time;
      Block Symplectic Gram–Schmidt step for X_{block} (Algorithm 2);
      end:=get time;
      $t_{ij} :=$ end $-$ start;
    end
    $a = (t_{i0} - t_{i1})/m_i$;
    $b[i] := (1/2)\, n^2 a + t_{i0} - a(h - m)$;
    for $j = 5:1$ **do**
      $A[ij] = m_i^{j-1}$;
    end
  end
  solve $A\mathbf{x} = \mathbf{b}$;
  $f(m) := x_1 m^4 + x_2 m^3 + x_3 m^2 + x_4 m + x_5$;
  solve $m := \min_{m\in[0,\frac{1}{2}N]} f(m)$;
end

4.1 Experiment one

Firstly, it is shown that the J-orthogonality of the SGS method is unstable and how our new re-orthogonalization condition is effective in addressing this. These numerical experiments were implemented in Matlab2013b and the test matrices were Hamiltonian matrices H with sizes 20×20, $40\times40, \ldots, 200\times200$ with random values. The CGS and the CSGS methods were employed to calculate orthogonality and J-orthogonality of the computed matrices by, respectively,

$$\|I - Q^T Q\|_2\,, \quad \|I - S^J S\|_2\,.. \tag{21}$$

Figures 1 and 2 show the numerical results. The results suggest that the J-orthogonality of the CSGS method is very unstable. For small size problems the accuracy of the J-orthogonality is approximately 10^{-10}. However, for a 200×200 Hamiltonian matrix the accuracy of the J-orthogonality is unacceptable. We have not identified the reason for this problem with large matrices, but it is possible that calculation errors are caused by the ESR increasing the norm of the computed vector.

Figure 2 illustrates the accuracy of the J-orthogonality of CSGS with re-orthogonalization condition (20), and it is much improved compared to Figure 1. It seems that the orthogonalization condition works for the CSGS method.

4.2 Experiment two

The effectiveness of the BSGS method is illustrated in this section. These numerical experiments were implemented in C language with double precision and the test matrices used were the Hamiltonian matrices $H_1 \in \mathbb{R}^{200\times200}$ and $H_2 \in \mathbb{R}^{2000\times2000}$ with random values. The SR factorization employed the BSGS, CSGS and MSGS methods and re-orthogonalization was implemented in each procedure.

Figure 1: J-Orthogonality of the CSGS method.

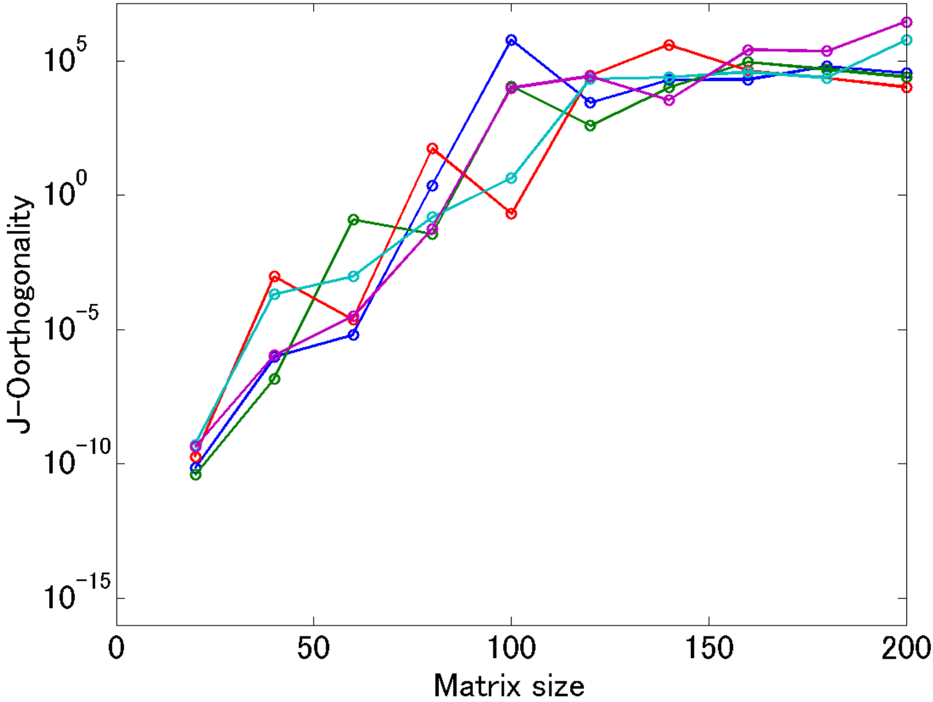

Table 1: Experiment 1.

Method	m	t_m	Accuracy
CSGS	2	0.104	8.70×10^{-6}
MSGS	2	0.107	4.65×10^{-6}
BSGS	10	0.039	2.80×10^{-6}

Tables 1 and 2 show results for the numerical experiments. In these tables 'Accuracy' refers to the calculation accuracy of the J-orthogonality by equation (21) and t_m is the computation time for block size m. BSGS-m refers to the BSGS method with our proposed method for determining block size (Algorithm 3).

Figure 2: J-Orthogonality of the CSGS with re-orthogonalization.

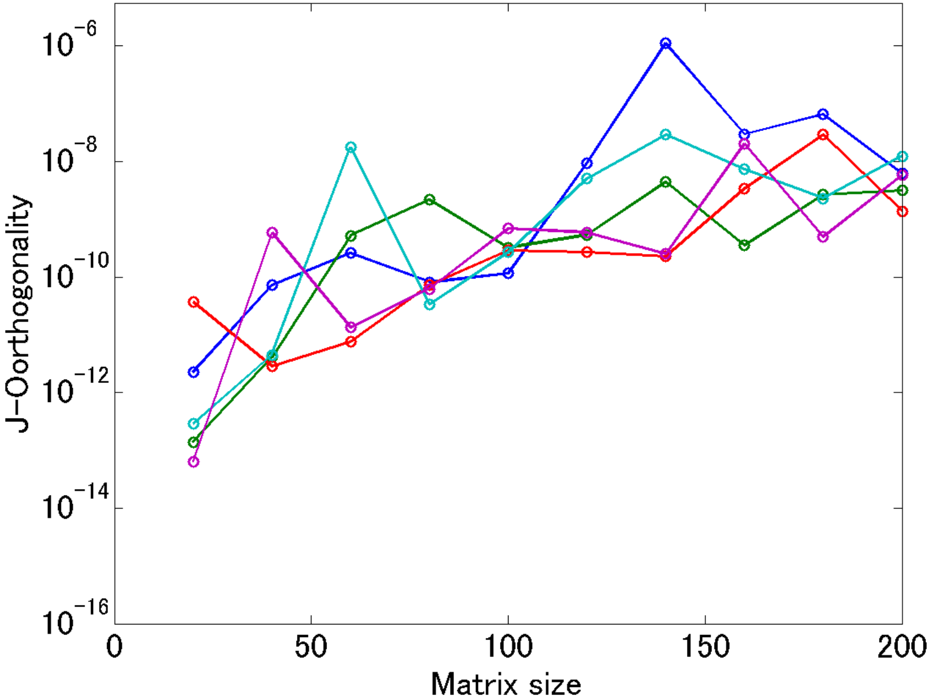

Table 2: Experiment 2.

Method	m	t_m	Accuracy
CSGS	2	21.24	1.55×10^{-4}
MSGS	2	21.40	1.03×10^{-4}
BSGS	40	2.50	3.74×10^{-5}
BSGS	100	3.21	7.14×10^{-5}
BSGS-m	72	2.72	4.48×10^{-5}

From Table 1 we see that the BSGS method is the fastest and has the highest accuracy in terms of J-orthogonality. This is because, after blocking X, we calculate the computation with Basic Linear Algebra Subprograms (BLAS). The accuracy of the BSGS method is significantly better than the accuracy of the CSGS method.

From Table 2 we again see that the BSGS method is the fastest and has the highest accuracy in terms of J-orthogonality. The BSGS method is approximately ten times faster than the CSGS and MSGS methods. The accuracy of the BSGS method is significantly better than that of the CSGS and MSGS methods. Since H_2 is larger than H_1, the SR factorization of matrix H_2 is more unstable than that of matrix H_1. The BSGS-m is not the fastest in Table 2, but the BSGS-m performed only 10% slower, more or less, than the fastest method.

5 Conclusion

The proposed BSGS method blocks the CSGS method to speed-up computation, and combines this with determining the optimal block size. Determining the block size is necessary because the computation time of the BSGS method changes significantly depending on block size.

Section 4 presented numerical experiments which show the effectiveness of the re-orthogonalization condition of the BSGS method. It is clear that this new condition worked for the SGS method. Our proposed method is much faster and more accurate than either the CSGS or the MSGS method. And in terms of determining block size, the BSGS method selects block size automatically.

In future studies it will be useful to analyze J-orthogonality and study how this proposed method works when dealing with a large scale problem.

References

[1] G. Ammar and V. Mehrmann. On Hamiltonian and symplectic Hessenberg forms. *Linear Algebra Appl.* 149:55–72, 1991. doi:10.1016/0024-3795(91)90325-Q C417

[2] P. Benner and H. Fassbender. An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem. *Linear Algebra Appl.* 263:75–111, 1997. doi:10.1016/S0024-3795(96)00524-1 C417

[3] A. Bunse-Gerstner and V. Mehrmann. A symplectic QR like algorithm for the solution of the real algebraic Riccati equation. *IEEE T. Automat. Contr.* 31:1104–1113, 1986. doi:10.1109/TAC.1986.1104186 C417

[4] G. Rünger and M. Schwind. Comparison of different parallel modified Gram–Schmidt algorithms, *Euro-Par 2005, Lecture Notes in Computer Science* 3648:826–836, 2005. doi:10.1007/11549468_90 C417

[5] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley, 1972. `http://dl.acm.org/citation.cfm?id=578807` C417

[6] C. Van Loan. A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix. *Linear Algebra Appl.*, 61:233–251, 1984. doi:10.1016/0024-3795(84)90034-X C417, C418

[7] Y. Matsuo and T. Nodera. The optimal block-size for the block Gram–Schmidt orthogonalization. *J. Sci. Tech.*, 49:569–584, 2011. C417, C418

[8] Y. Matsuo and T. Nodera. An Efficient Implementation of the Block Gram–Schmidt Method. *CTAC2012, ANZIAM J.*, 54:C394–409, 2013. `http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6327` C417, C423

[9] A. Salam. On theoretical and numerical aspects of symplectic Gram–Schmidt-like algorithms, *Numer. Algorithms* 39:437–462, 2005. doi:10.1007/s11075-005-0963-2 C418, C423

[10] A. Salam and E. Al-Aidarous. Equivalence between modified symplectic Gram–Schmidt and Householder SR algorithms. *BIT Numer. Math.*, 54:283–302, 2014. doi:10.1007/s10543-013-0441-5 C418, C419

[11] S. J. Leon, A. Björck and W. Gander. Gram–Schmidt orthogonalization: 100 years and more. *Numer. Linear Algebra Appl.* 20:492–532, 2013. doi:10.1002/nla.1839 C417

[12] G. W. Stewart. Block Gram–Schmidt orthogonalization, *SIAM J. Sci. Comput.*, 31:761–775, 2008. doi:10.1137/070682563 C417, C418, C421

Author addresses

1. **Yoichi Matsuo**, School of Fundamental Science and Technology, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan.
   mailto:matsuo@math.keio.ac.jp
2. **Takashi Nodera**, Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan.
   mailto:nodera@math.keio.ac.jp