BULLETIN *of the*
Malaysian Mathematical
Sciences Society
http://math.usm.my/bulletin

Bull. Malays. Math. Sci. Soc. (2) **34**(2) *(2011), 307–318*

Inverses of Block Tridiagonal Matrices and Rounding Errors

[1]Chi-Ye Wu, [2]Ting-Zhu Huang, [3]Liang Li and [4]Xiao-Guang Lv

[1]Jinan University, Shenzhen, Guangdong, 518053, P. R. China
[1,2,3,4] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054, P. R. China
[1]chyewu@jnu.edu.cn, [2]tingzhuhuang@126.com,
[3]plum_liliang@uestc.edu.cn, [4]little006@126.com

Abstract. Based on URV-decomposition in Stewart [An updating algorithm for subspace tracking, *IEEE Trans. Signal Processing*, 40 (1992): 1535–1541] and the result of Mehrmann [Divide and conquer methods for block tridiagonal systems, *Parallel Comput.*, 19 (1993): 257–279], inverses of block tridiagonal matrices are presented. The computational complexity of the proposed algorithm is less than that of the Block Gaussian-Jordan Elimination method when the orders of the matrices are not less than 100. Expressions for the rounding errors incurred during the process of the computation of the inverses of block tridiagonal matrices are also considered. Moreover, from the experiment, it shows that the norms of the errors generated from the Block Gaussian-Jordan Elimination method are larger than those of the proposed algorithm.

2010 Mathematics Subject Classification: 65F05, 65G50, 65Y20

Keywords and phrases: Block tridiagonal matrices, inverse, divide-and-conquer algorithm, block Gaussian-Jordan elimination, rounding error.

1. Introduction

Tridiagonal matrices are connected with different areas of science and engineering, including telecommunication system analyses [7] and finite difference methods for solving partial differential equations [2, 14, 10]. Large tridiagonal systems appear in many applications, such as finite elements, difference schemes to differential equations, power distribution systems, etc.

In many of these areas, inverses of tridiagonal matrices are necessary. Efficient algorithms [4], indirect formulae [1, 12, 16], and direct expressions in some special cases [2, 10] for such inversions are known. However, the investigation on inverses of block tridiagonal matrices is relatively few. The Block Gaussian-Jordan Elimination method is one of them. But the computational complexity is $O(\frac{1}{3}n^3)$. Having

Communicated by Norhashidah Hj. Mohd. Ali.
Received: June 2, 2009; *Revised:* August 4, 2010.

referred to the recent publications, Meurant [12] presented the results on inverses of symmetric block tridiagonal matrices. Based on the twisted block decompositions of block tridiagonal matrices, the explicit expressions of the block elements of matrices inverses were presented in Ran and Huang [13]. In this paper, based on the divide-and-conquer algorithm and the URV-decomposition and from the result of Volker Mehrmann [11], inverses of block tridiagonal matrices are presented. The computational complexity is $O(\frac{1}{4^l}n^3)$ which is less than that of the Block Gaussian-Jordan Elimination method, where l is the timings of the divide-and-conquer algorithm applied for the inverse and n is very large.

To most numerical analysts, matrix inversion is a sin. Not only is the inverse approach three times more expensive, but also it is much less stable. However, there are situations, such as statistics where the inverse can convey important statistical information [8] and numerical integrations arising in superconductivity computations [5], in which a matrix inverse must be computed. Moreover, methods for a matrix inversion display a wide variety of stability properties. A rounding error analysis was presented for a divide-and-conquer algorithm to solve linear systems with block Hessenberg matrices in [9] and was stable for block diagonally dominant matrices and for M-matrices. In this paper we also establish rounding errors incurred during the computation of inverses of block tridiagonal matrices. Moreover, from the results of the experiment, it shows that the norms of the errors generated from the proposed algorithm are less than those of the Block Gaussian-Jordan Elimination method.

Unless otherwise stated, throughout our analysis we will use the 2-norm. Its main advantage is that the norm of an orthogonal matrix is one. The computed matrices, vectors and scalars will be identified by a hat over the symbol.

2. Inverses of block tridiagonal matrices

Consider a nonsingular block tridiagonal matrix

$$A = \text{tridiag}(C, D, B)_{s\times s} \in \mathbb{R}^{n\times n}, \tag{2.1}$$

where $s > 1$,

$$C = [C_2, \cdots, C_s]^T, \quad D = [D_1, \cdots, D_s]^T, \quad B = [B_1, \cdots, B_{s-1}]^T,$$

$D_i \in \mathbb{R}^{k_i\times k_i}$ $(i = 1, 2, \ldots, s)$ are arbitrary matrices, the off-diagonal blocks $C_i \in \mathbb{R}^{k_i\times k_{i-1}}(i = 2, 3, \ldots, s)$ and $B_i \in \mathbb{R}^{k_i\times k_{i+1}}(i = 1, 2, \ldots, s-1)$ have rank r, and k_i satisfy $1 \le k_i < n$ and $\sum_{i=1}^{s} k_i = n$.

If $s > 1$ we can partition the matrix A as follows:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}. \tag{2.2}$$

The submatrix A_{11} contains the first k diagonal blocks of A and A_{22} contains the last $s-k$ diagonal blocks of A, where the dimensions $n_1 = \sum_{i=1}^{k} k_i$ and $n_2 = \sum_{i=k+1}^{s} k_i$. Note that B_k and C_{k+1} are the only nonzero blocks in A_{12} and A_{21}, respectively. Partitioning the matrix A in (2.1), we first obtain the following lemma.

Lemma 2.1. *Let the matrix A partitioned in* (2.1) *be the same as that in* (2.2). *Then*

$$A_1 = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} = A - EA_{21}F^T$$

and

$$A_2 = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} = A_1 - FA_{12}E^T,$$

where E and F are the last n_2 columns of the n-by-n identity matrix and the first n_1 columns of the n-by-n identity matrix, respectively.

Proof. Let E and F be the last n_2 columns of the n-by-n identity matrix and the first n_1 columns of the n-by-n identity matrix, respectively, that is,

$$E = \begin{pmatrix} 0 & I_{n_2} \end{pmatrix}^T, \quad F = \begin{pmatrix} I_{n_1} & 0 \end{pmatrix}^T.$$

Then

$$\begin{aligned} EA_{21}F^T &= \begin{pmatrix} 0 \\ I_{n_2} \end{pmatrix}_{n\times n_2} \begin{pmatrix} 0 & C_{k+1} \\ 0 & 0 \end{pmatrix}_{n_2\times n_1} \begin{pmatrix} I_{n_1} & 0 \end{pmatrix}_{n_1\times n} \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & C_{k+1} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{n\times n} \\ &= \begin{pmatrix} 0 & 0 \\ A_{21} & 0 \end{pmatrix}_{n\times n}. \end{aligned} \tag{2.3}$$

From (2.2) and (2.3), we have

$$A_1 = A - EA_{21}F^T.$$

The following proof with respect to the relationship between A_1 and A_2 is analogous, so we omit it. Hence

$$A_2 = A_1 - FA_{12}E^T.$$

The proof is completed. ∎

Now we recall the well-known Sherman-Morrison-Woodbury formula concerning the inverse of $A + XY^T$ to obtain the inverse of A in (2.1).

Lemma 2.2. [3]. *Let $A \in \mathbf{R}^{n\times n}$ and $I + Y^TA^{-1}X$ be nonsingular, and X, $Y \in \mathbf{R}^{n\times k}$. Then*

$$(A + XY^T)^{-1} = A^{-1} - A^{-1}X(I + Y^TA^{-1}X)^{-1}Y^TA^{-1}.$$

Many problems in certain signal processing applications require the computation of an approximate null space of an $n \times p$ matrix whose rows represent samples of a signal. The usual tool for doing it is the singular value decomposition. However, it has the drawback that it requires $O(p^3)$ operations. Therefore, Stewart [15] shows that a different decomposition, called the URV-decomposition, is equally effective in exhibiting the null space and can be updated in $O(p^2)$ time. The following is the URV-decomposition for the matrices A_{21} and A_{12}.

Lemma 2.3. [15]. *Let the matrix A_{21} in (2.2) be of rank r. Then there are orthogonal matrices $U_2 \in \mathbf{R}^{n_2\times r}$ with k_{+1} nonzero rows and $V_2 \in \mathbf{R}^{n_1\times r}$ with k_k nonzero rows such that*

$$A_{21} = U_2R_2V_2^T, \tag{2.4}$$

where R_2 is a square $r \times r$ matrix.

For the matrix A_{12} in (2.2), there are also orthogonal matrices $U_1 \in \mathbf{R}^{n_1 \times r}$ with k_k nonzero rows and $V_1 \in \mathbf{R}^{n_2 \times r}$ with k_{k+1} nonzero rows such that

$$A_{12} = U_1 R_1 V_1^T, \tag{2.5}$$

where R_1 is a square $r \times r$ matrix.

From Lemma 2.1 and Lemma 2.2, we have the following lemma.

Lemma 2.4. [11]. *Let the matrix A be in* (2.1) *and $I + A_{21}F^T A_1^{-1} E$ be nonsingular. Then the inverse formula of the matrix A is as follows:*

$$A^{-1} = A_1^{-1} - A_1^{-1} E (I + A_{21} F^T A_1^{-1} E)^{-1} A_{21} F^T A_1^{-1},$$

where

$$A_1^{-1} = A_2^{-1} - A_2^{-1} F A_{12} E^T A_2^{-1}$$

and $A_2^{-1} = \mathrm{diag}(A_{11}^{-1}, A_{22}^{-1})$.

From Sherman-Morrison-Woodbury formula and URV-decomposition, the inverse of block tridiagonal matrix A in (2.1) is presented when the timings of the divide-and-conquer algorithm applied is one. With the help of the URV-decomposition [15] and Lemma 2.4, we have the following theorem.

Theorem 2.1. *For the matrix A in* (2.1)*, if $I + R_2 V_2^T F^T A_1^{-1} E U_2$ is nonsingular, then the inverse formula of the matrix A is as follows:*

$$A^{-1} = A_1^{-1} - A_1^{-1} E U_2 (I + R_2 V_2^T F^T A_1^{-1} E U_2)^{-1} R_2 V_2^T F^T A_1^{-1}, \tag{2.6}$$

where

$$A_1^{-1} = A_2^{-1} - A_2^{-1} F U_1 R_1 V_1^T E^T A_2^{-1} \tag{2.7}$$

and $A_2^{-1} = \mathrm{diag}(A_{11}^{-1}, A_{22}^{-1})$.

Proof. From (2.4) and (2.5), we have

$$A = A_1 + E U_2 R_2 V_2^T F^T = A_1 + (E U_2)(R_2 V_2^T F^T),$$
$$A_1 = A_2 + F U_1 R_1 V_1^T E^T = A_2 + (F U_1)(R_1 V_1^T E^T).$$

Applying the Sherman-Morrison-Woodbury formula gives

$$A^{-1} = A_1^{-1} - A_1^{-1} E U_2 (I + R_2 V_2^T F^T A_1^{-1} E U_2)^{-1} R_2 V_2^T F^T A_1^{-1},$$
$$A_1^{-1} = A_2^{-1} - A_2^{-1} F U_1 (I + R_1 V_1^T E^T A_2^{-1} F U_1)^{-1} R_1 V_1^T E^T A_2^{-1}.$$

Using the multiplication of matrices, we have

$$\begin{aligned} R_1 V_1^T E^T A_2^{-1} F U_1 &= (R_1)_{r\times r} (V_1^T)_{r\times n_2} \begin{pmatrix} 0 & I_{n_2} \end{pmatrix}_{n_2 \times n} \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & A_{22}^{-1} \end{pmatrix}_{n\times n} \\ &\quad \begin{pmatrix} I_{n_1} \\ 0 \end{pmatrix}_{n\times n_1} (U_1)_{n_1 \times r} \\ &= \begin{pmatrix} 0 & (R_1 V_1^T)_{r\times n_2} \end{pmatrix}_{r\times n} \begin{pmatrix} (A_{11}^{-1} U)_{n_1 \times r} \\ 0 \end{pmatrix}_{n\times r} \\ &= 0_{r\times r}. \end{aligned}$$

Therefore,

$$A_1^{-1} = A_2^{-1} - A_2^{-1} F U_1 R_1 V_1^T E^T A_2^{-1}.$$

The proof is completed. ∎

In fact, the above proof can be simplified, that is, substituting the two formulas in Lemma 2.3 into Lemma 2.4 and reorganizing them, the theorem follows.

As mentioned above, we obtain the following theorem when B_i and C_j are zero matrices for all $i \neq k$ and $j \neq k+1$, respectively.

Theorem 2.2. *Let B_i and C_j be zero matrices for all $i \neq k$ and $j \neq k+1$. Then*

$$A^{-1} = A_1^{-1} - A_1^{-1}EU_2(I + R_2V_2^TF^TA_1^{-1}EU_2)^{-1}R_2V_2^TF^TA_1^{-1},$$

where

$$A_1^{-1} = \mathrm{diag}(D_1^{-1}, \ldots, D_s^{-1}) - \mathrm{diag}(D_1^{-1}, \ldots, D_s^{-1})FU_1R_1V_1E^T\mathrm{diag}(D_1^{-1}, \ldots, D_s^{-1}).$$

Proof. Since B_i and C_j be zero matrices for all $i \neq k$ and $j \neq k+1$, then A_{11} and A_{22} are both block diagonal matrices, that is,

$$A_2^{-1} = \mathrm{diag}(D_1^{-1}, \ldots, D_s^{-1}).$$

Applying Theorem 2.1 gives

$$A^{-1} = A_1^{-1} - A_1^{-1}EU_2(I + R_2V_2^TF^TA_1^{-1}EU_2)^{-1}R_2V_2^TF^TA_1^{-1}. \quad \blacksquare$$

Using the divide-and-conquer algorithm, we can obtain two smaller dimensions matrices. Hence inverses of the smaller block tridiagonal matrices are simpler than those of original matrices. This step requires the inverses with the matrices A_{11} and A_{22}, which can be solved recursively by the same divide-and-conquer algorithm. An added advantage of the presented algorithm is much more conspicuous when we compute inverses of large block tridiagonal matrices. Furthermore, note that this algorithm may be applied to parallel computation.

3. Rounding error analysis

Throughout, we use the "standard model" of floating-point arithmetic in which the evaluation of an expression in floating-point arithmetic is denoted by $fl(\cdot)$, with

$$fl(a \ o \ b) = (a \ o \ b)(1+\delta), \quad |\delta| \leq u, \ o = +, -, *, /$$

(see, for example, Higham [6]). Here u is the unit rounding off associated with the particular machine being used.

In the case of the addition of two matrices, we have

$$fl(A+B) = A + B + W_1,$$

where

$$\|W_1\| \leq \eta_1\|A+B\|.$$

The quantity η_1 is on the order of the unit round-off u and slowly increases with the size of the matrices A and B.

If we multiply two matrices in floating-point arithmetic, we obtain

$$fl(AB) = AB + W_2,$$

where

$$\|W_2\| \leq \eta_2\|A\|\|B\|.$$

The quantity η_2 is a small multiple of the unit round-off and slowly grows with the size of the matrices A and B (see, for example, Wilkinson [17]).

In this section we will give expressions for the rounding errors incurred during the process of the computation of the inverse of A in (2.1). Consider firstly the rounding error of the inverse of A_1. Then we further consider that of the inverse of A.

3.1. Calculation of the inverse of A_1

The computed approximation $\hat{U}_i$, $\hat{R}_i$ and $\hat{V}_i$ can be described by

$$A_{12} = \hat{U}_1\hat{R}_1\hat{V}_1^T + \Delta A_{12}, \quad A_{21} = \hat{U}_2\hat{R}_2\hat{V}_2^T + \Delta A_{21}, \tag{3.1}$$

where

$$\|\Delta A_{12}\| \le 4\eta_3\|A_{12}\|, \quad \|\Delta A_{21}\| \le 4\eta_3\|A_{21}\|. \tag{3.2}$$

The quantity η_3 is on the order of the unit round-off and slowly grows with the size of the matrices A_{12} and A_{21}, respectively. The matrices $\hat{U}_1$, $\hat{U}_2$, $\hat{V}_1$ and $\hat{V}_2$ are nearly orthogonal, then

$$\|\hat{U}_i\| \le 1 + \eta_3, \quad \|\hat{V}_i\| \le 1 + \eta_3, \quad i = 1, 2. \tag{3.3}$$

The matrices G and H can be expressed as

$$G = \hat{A}_2^{-1}F\hat{U}_1\hat{R}_1\hat{V}_1^T + \Delta G, \tag{3.4}$$

$$H = GE^T\hat{A}_2^{-1} + \Delta H, \tag{3.5}$$

where

$$\begin{aligned} \|\Delta G\| &\le \eta_2\|\hat{A}_2^{-1}F\|\|\hat{U}_1\hat{R}_1\hat{V}_1^T\| \\ &\le \eta_2\|\hat{A}_2^{-1}\|\|A_{12} - \Delta A_{12}\| \\ &\le \eta_2(1 + 4\eta_3)\|\hat{A}_2^{-1}\|\|A_{12}\|, \end{aligned} \tag{3.6}$$

$$\begin{aligned} \|\Delta H\| &\le \eta_2\|G\|\|E^T\hat{A}_2^{-1}\| \\ &\le \eta_2\|G\|\|\hat{A}_2^{-1}\|. \end{aligned} \tag{3.7}$$

From (3.1) and (3.4), we have

$$G = \hat{A}_2^{-1}F(A_{12} - \Delta A_{12}) + \Delta G = \hat{A}_2^{-1}FA_{12} - \hat{A}_2^{-1}F\Delta A_{12} + \Delta G. \tag{3.8}$$

Taking the norm of both sides gives

$$\|G\| \le \|\hat{A}_2^{-1}FA_{12}\| + \|\hat{A}_2^{-1}F\Delta A_{12}\| + \|\Delta G\|. \tag{3.9}$$

Using inequalities (3.2), (3.6) and (3.9), we get

$$\begin{aligned} \|G\| &\le (1 + 4\eta_3)\|\hat{A}_2^{-1}\|\|A_{12}\| + \eta_2(1 + 4\eta_3)\|\hat{A}_2^{-1}\|\|A_{12}\| \\ &= (1 + 4\eta_3)(1 + \eta_2)\|\hat{A}_2^{-1}\|\|A_{12}\|. \end{aligned} \tag{3.10}$$

From (3.7) and (3.10), we have

$$\|\Delta H\| \le \eta_2(1 + 4\eta_3)(1 + \eta_2)\|\hat{A}_2^{-1}\|^2\|A_{12}\|. \tag{3.11}$$

From (3.5), (3.10) and (3.11), it follows that

$$\begin{aligned} \|H\| &\le \|G\|\|E^T\hat{A}_2^{-1}\| + \|\Delta H\| \\ &\le (1 + 4\eta_3)(1 + \eta_2)\|\hat{A}_2^{-1}\|^2\|A_{12}\| + \eta_2(1 + 4\eta_3)(1 + \eta_2)\|\hat{A}_2^{-1}\|^2\|A_{12}\| \end{aligned} \tag{3.12}$$

$$= (1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|^2\|A_{12}\|.$$

Using equations (2.7) and (3.5), we obtain

$$\hat{A}_1^{-1} = \hat{A}_2^{-1} - H + \Delta\hat{A}_1^{-1}, \tag{3.13}$$

where

$$\begin{aligned}\|\Delta\hat{A}_1^{-1}\| &\le \eta_1\|\hat{A}_2^{-1} - H\| \\ &\le \eta_1(\|\hat{A}_2^{-1}\| + \|H\|).\end{aligned} \tag{3.14}$$

From (3.12), (3.13) and (3.14), we obtain

$$\begin{aligned}\|\hat{A}_1^{-1}\| &\le (1+\eta_1)(\|\hat{A}_2^{-1}\| + \|H\|) \\ &\le (1+\eta_1)(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\|.\end{aligned} \tag{3.15}$$

Therefore,

$$\begin{aligned}\|\Delta\hat{A}_1^{-1}\| &\le \eta_1(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\| \\ &= \eta_1(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\| + O(u^2).\end{aligned}$$

From the above proof, we have the following proposition.

Proposition 3.1. *Let A in (2.1) be nonsingular. Then the norm of the rounding error incurred during the process of the computation of the inverse of the matrix A_1 satisfies*

$$\|\Delta\hat{A}_1^{-1}\| \le \eta_1(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\| + O(u^2),$$

where η_i for all $1 \le i \le 3$ are on the order of the unit round-off u and slowly grow with the size of matrices.

3.2. Calculation of the inverse of A

In this subsection, the norm of the rounding error incurred during the process of the computation of the inverse of the matrix A is presented when the timings of the divide-and-conquer algorithm applied is one.

Proposition 3.2. *Let A in (2.1) be nonsingular. Then the rounding error of the inverse of the matrix A*

$$\|\Delta\hat{A}^{-1}\| \le \eta_1(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)(1+1.01\|\hat{A}^{-1}EA_{21}\|)\|\hat{A}_2^{-1}\| + O(u^2),$$

where η_i for all $1 \le i \le 3$ are on the order of the unit round-off u and slowly grow with the size of matrices.

Proof. In order to avoid the multiple evaluation of the same expressions, we introduce the following intermediate quantities:

$$P := \hat{A}_1^{-1}E\hat{U}_2, \quad Q := \hat{R}_2\hat{V}_2^T F^T P,$$

$$K := I + Q, \qquad L := K^{-1}\hat{R}_2, \quad S := PL.$$

The matrix S can be expressed as

$$S = PL + \Delta S. \tag{3.16}$$

Matt and Stewart give the following results on the calculation of the matrix S in [9].

$$\|\Delta S\| \le 1.01^2\eta_2\|\hat{A}_1^{-1}\|\|\hat{A}_1 - \hat{A}_1\hat{A}^{-1}\hat{A}_1\|, \quad \|S\| \le 1.01\|\hat{A}^{-1}EA_{21}\|, \tag{3.17}$$

where 1.01 is a factor to simplify our bounds. The matrix T can also be expressed as

$$T = S\hat{V}_2^T F^T \hat{A}_1^{-1} + \Delta T, \tag{3.18}$$

where

$$\|\Delta T\| \le \eta_2 \|S\| \|\hat{V}_2^T F^T \hat{A}_1^{-1}\|. \tag{3.19}$$

From (3.3), (3.15), (3.17) and (3.19), we have

$$\begin{aligned}\|\Delta T\| &\le 1.01\eta_2(1+\eta_3)\|\hat{A}^{-1}EA_{21}\|\|\hat{A}_1^{-1}\| \\ &\le 1.01\eta_2(1+\eta_3)(1+\eta_1) \\ &\quad \times (1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\|\|\hat{A}^{-1}EA_{21}\|. \end{aligned} \tag{3.20}$$

Therefore,

$$\begin{aligned}\|T\| &\le 1.01(1+\eta_2)(1+\eta_3)(1+\eta_1) \\ &\quad \times (1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|)\|\hat{A}_2^{-1}\|\|\hat{A}^{-1}EA_{21}\|. \end{aligned} \tag{3.21}$$

From equations (2.6) and (3.18), we have

$$\hat{A}^{-1} = \hat{A}_1^{-1} - T + \Delta\hat{A}^{-1}. \tag{3.22}$$

By inequalities (3.15) and (3.21), we obtain

$$\begin{aligned}\|\Delta\hat{A}^{-1}\| &\le \eta_1\|\hat{A}_1^{-1} - T\| \\ &\le \eta_1(\|\hat{A}_1^{-1}\| + \|T\|) \\ &\le \eta_1\Bigg[(1+\eta_1)\bigg(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|\bigg)\|\hat{A}_2^{-1}\| \\ &\quad + 1.01(1+\eta_2)(1+\eta_3)(1+\eta_1) \\ &\quad \times \bigg(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|\bigg)\|\hat{A}_2^{-1}\|\|\hat{A}^{-1}EA_{21}\|\Bigg] \\ &= \eta_1(1+\eta_1)\bigg(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|\bigg) \\ &\quad \times \bigg(1+1.01(1+\eta_2)(1+\eta_3)\|\hat{A}^{-1}EA_{21}\|\bigg)\|\hat{A}_2^{-1}\| \\ &= \eta_1(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)(1+1.01\|\hat{A}^{-1}EA_{21}\|)\|\hat{A}_2^{-1}\| + O(u^2). \end{aligned} \tag{3.23}$$

From (3.22) and (3.23), it follows that

$$\begin{aligned}\|\hat{A}^{-1}\| &\le (1+\eta_1)(\|\hat{A}_1^{-1}\| + \|T\|) \\ &\le (1+\eta_1)^2\bigg(1+(1+4\eta_3)(1+\eta_2)^2\|\hat{A}_2^{-1}\|\|A_{12}\|\bigg) \\ &\quad \times \bigg(1+1.01(1+\eta_2)(1+\eta_3)\|\hat{A}^{-1}EA_{21}\|\bigg)\|\hat{A}_2^{-1}\| \\ &= (1+\|\hat{A}_2^{-1}\|\|A_{12}\|)(1+1.01\|\hat{A}^{-1}EA_{21}\|)\|\hat{A}_2^{-1}\| \\ &\quad + 2(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)(1+1.01\|\hat{A}^{-1}EA_{21}\|)\|\hat{A}_2^{-1}\|\eta_1 \end{aligned}$$

$$
\begin{aligned}
&+\Big(2\|\hat{A}_2^{-1}\|\|A_{12}\|(1+1.01\|\hat{A}^{-1}EA_{21}\|)\\
&+\|\hat{A}^{-1}EA_{21}\|(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)\Big)\|\hat{A}_2^{-1}\|\eta_2\\
&+\Big(4\|\hat{A}_2^{-1}\|\|A_{12}\|(1+1.01\|\hat{A}^{-1}EA_{21}\|)\\
&+\|\hat{A}^{-1}EA_{21}\|(1+\|\hat{A}_2^{-1}\|\|A_{12}\|)\Big)\|\hat{A}_2^{-1}\|\eta_3+O(u^2). \qquad \blacksquare
\end{aligned}
$$

For large block tridiagonal matrices, using recursively the same divide-and-conquer algorithm for the inverses of the matrices A_{11} and A_{22}, the errors incurred during the process of the computation of the inverses can also be presented. Here we ignore it.

4. Comparison on computational complexity and numerical experiment

Unless otherwise stated, in this section we denote $k_i = k_{i+1} = m$ and $n_1 = n_2$.

For the computational complexity, we only consider the multiplication and division. For a matrix of order n, we use the Block Gaussian-Jordan Elimination method for inverting it. It is easy to know the computational complexity of the Block Gaussian-Jordan Elimination method is $(\frac{1}{3}s^3+\frac{3}{2}s^2+\frac{7}{6}s-2)m^3$ $(n=s\times m)$.

On the other hand, applying the divide-and-conquer algorithm to inverses of block tridiagonal matrices, the computational complexity of the proposed algorithm is $\frac{1}{4}n^3+2n^2r+(m^2+3\frac{r^2}{2})n+r^3+O(r^2)$, where $O(r^2)$ is the computational complexity of URV-decomposition. For large block tridiagonal matrices, using recursively the same divide-and-conquer algorithm for the inverses of the matrices A_{11} and A_{22}, the computational complexity is as follows:

$$
\begin{aligned}
&2^l(\tfrac{n}{2^l})^3+2n^2r+(m^2+3\tfrac{r^2}{2})n+r^3\\
&\quad+2(\tfrac{n}{2})^2 2r+2(m^2+3\tfrac{r^2}{2})\tfrac{n}{2}+r^3\\
&\quad+2(\tfrac{n}{2^2})^2 2^2 r+2^2(m^2+3\tfrac{r^2}{2})\tfrac{n}{2^2}+r^3\\
&\quad\cdots\cdots\\
&\quad+2(\tfrac{n}{2^{l-1}})^2 2^{l-1}r+2^{l-1}(m^2+3\tfrac{r^2}{2})\tfrac{n}{2^{l-1}}+r^3+O(r^2)\\
&=\tfrac{n^3}{4^l}+(4-\tfrac{1}{2^{l-2}})n^2r+(m^2+3\tfrac{r^2}{2})ln+lr^3+O(r^2),
\end{aligned}
$$

where l is the timings of the divide-and-conquer algorithm applied for the inverse.

Comparing the computational complexity of the Block Gaussian-Jordan Elimination method with that of the proposed algorithm, especially for large block tridiagonal matrices, it is conspicuous that the computational complexity of the proposed algorithm is less than that of the Block Gaussian-Jordan Elimination method. See Table 1.

Table 1. Comparison of the computational complexity

Block Gaussian-Jordan Elimination	$(\frac{1}{3}s^3+\frac{3}{2}s^2+\frac{7}{6}s-2)m^3$
Algorithm in this paper	$\frac{n^3}{4^l}+(4-\frac{1}{2^{l-2}})n^2r+(m^2+3\frac{r^2}{2})ln+lr^3+O(r^2)$

Now we use a numerical test to illustrate our results analyses. The test is performed on a Lenovo PC, with 1Gb memory and a 3GHz Pentium(R) D CPU.

Example 4.1. We consider a block tridiagonal matrix $A = \text{tridiag}(-I, D, -I)$ generated from the discretization of partial differential equation $-\Delta u = f$, where $D = \text{tridiag}(-1, 4, -1)_{m\times m}$. Then the norms $\|I-A\hat{A}^{-1}\|$, $\|I-\hat{A}^{-1}A\|$, and the computational complexity in this test are considered, where $\hat{A}^{-1}$ and the computational complexity are generated from the Block Gaussian-Jordan Elimination method and the proposed algorithm, respectively. See Tables 2 and 3.

Table 2. Comparison of the errors

	Block Gaussian-Jordan Elimination		Algorithm in this paper	
Size	$\|I-A\hat{A}^{-1}\|$	$\|I-\hat{A}^{-1}A\|$	$\|I-A\hat{A}^{-1}\|$	$\|I-\hat{A}^{-1}A\|$
64×64	5.6826e-015	2.8974e-015	3.5562e-015	2.1641e-015
256×256	3.8003e-014	1.2996e-014	1.1563e-014	9.2903e-015
576×576	1.1127e-013	3.1285e-014	2.9638e-014	2.6837e-014
1024×1024	2.0945e-013	6.4845e-014	5.5750e-014	4.3897e-014
1600×1600	4.1792e-013	1.0672e-013	9.1734e-014	7.8641e-014

Table 3. Comparison of the computational complexities

Size	Block Gaussian-Jordan Elimination	Algorithm in this paper
64×64	140,290	156,672
100×100	493,000	482,000
256×256	7,233,536	4,792,320
576×576	76,004,352	35,748,864
1024×1024	409,403,392	146,866,176
1600×1600	1,521,800,000	445,568,000

From Tables 2 and 3, we have the following conclusions. Firstly, it shows that the computational complexity of the Block Gaussian-Jordan Elimination method is less than that of the proposed algorithm when the orders of matrices are not more than 81. However, as the orders of matrices gradually increase, the computational complexity of the Block Gaussian-Jordan Elimination method is obviously more than that of the proposed algorithm. Secondly, the norms of the errors generated from

the Block Gaussian-Jordan Elimination method are also always larger than those of the proposed algorithm.

From Tables 1 and 3, we find that the computational complexity in practice is always less than that in theory because of the sparsity of the submatrices, where the computational complexity in practice is presented by the accumulation of the number of floating-point arithmetic during the process.

5. Conclusions

Applying the similar method to [9], the URV-decomposition and the result of Mehrmann [11], we have established inverses of block tridiagonal matrices and expressions for the rounding errors incurred during the process of the computation of the inverse of the matrix A in (2.1). When we use recursively the same divide-and-conquer algorithm for the inverses of the matrices A_{11} and A_{22}, the computational complexity is $O(\frac{n^3}{4^l})$, where l is the timings of the divide-and-conquer algorithm applied to the inverse. However, the computational complexity of the Block Gaussian-Jordan Elimination method is $O(\frac{n^3}{3})$. Therefore the computational complexity of the proposed algorithm is less than that of the Block Gaussian-Jordan Elimination method when the orders of the matrices are very large. Our conclusions have been verified by numerical tests that we have conducted. This algorithm may be applied to parallel computation and this will be addressed in a future study.

Acknowledgement. The authors would like to thank the anonymous referee and the editor Prof. Dato' Rosihan M. Ali very much for their helpful comments and suggestions that have helped to improve the presentation of this paper. This research was supported by NSFC (60973015) and Sichuan Province Sci. & Tech. Research Project (2009SPT-1, 2009GZ0004).

References

[1] W. W. Barrett, A theorem on inverses of tridiagonal matrices, *Linear Algebra Appl.* **27** (1979), 211–217.

[2] C. F. Fischer and R. A. Usmani, Properties of some tridiagonal matrices and their application to boundary value problems, *SIAM J. Numer. Anal.* **6** (1969), 127–142.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, second edition, Johns Hopkins Series in the Mathematical Sciences, 3, Johns Hopkins Univ. Press, Baltimore, MD, 1989.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, third edition, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins Univ. Press, Baltimore, MD, 1996.

[5] M. T. Heath, G. A. Geist and J. B. Drake, *Early experience with the Intel iPSC/860 at Oak Ridge National Laboratory, Report ORNL/TM-11655*, Oak Ridge, TN, USA, 1990.

[6] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, second edition, SIAM, Philadelphia, PA, 2002.

[7] L. Lu and W. Sun, The minimal eigenvalues of a class of block-tridiagonal matrices, *IEEE Trans. Inform. Theory* **43** (1997), no. 2, 787–791.

[8] J. H. Maindonald, *Statistical Computation*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley, New York, 1984.

[9] U. von Matt and G. W. Stewart, Rounding errors in solving block Hessenberg systems, *Math. Comp.* **65** (1996), no. 213, 115–135.

[10] R. M. M. Mattheij and M. D. Smooke, Estimates for the inverse of tridiagonal matrices arising in boundary value problems, *Linear Algebra Appl.* **73** (1986), 33–57.

[11] V. Mehrmann, Divide and conquer methods for block tridiagonal systems, *Parallel Comput.* **19** (1993), no. 3, 257–279.
[12] G. Meurant, A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, *SIAM J. Matrix Anal. Appl.* **13** (1992), no. 3, 707–728.
[13] R.-S. Ran and T.-Z. Huang, The inverses of block tridiagonal matrices, *Appl. Math. Comput.* **179** (2006), no. 1, 243–247.
[14] G. D. Smith, *Numerical Solution of Partial Differential Equations*, second edition, Oxford Univ. Press, New York, 1978.
[15] G. W. Stewart, An updating algorithm for subspace tracking, *IEEE Trans. Signal Processing*, **40** (1992), 1535–1541.
[16] T. Yamamoto and Y. Ikebe, Inversion of band matrices, *Linear Algebra Appl.* **24** (1979), 105–111.
[17] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.