ALGORITHMS FOR CENTROSYMMETRIC AND SKEW-CENTROSYMMETRIC MATRICES

Iyad T. Abu-Jeib

Abstract. We present a simple algorithm that reduces the time complexity of solving the linear system $Gx = b$, where G is a centrosymmetric/skew-centrosymmetric matrix. We also reduce the time complexity of solving some complex linear systems. We propose efficient methods for multiplying centrosymmetric/skew-centrosymmetric matrices.

1. Introduction. Melman [7] proposed an efficient algorithm for computing the product Gv, where G is a symmetric centrosymmetric matrix and v is a vector. In this paper, we present a simple efficient algorithm for solving $Gx = b$, where G is a centrosymmetric/skew-centrosymmetric matrix and b is a vector. In our algorithm G does not have to be symmetric or skew-symmetric. Some complex linear systems can be converted to real systems and then solved by using our algorithm. Finally, we propose efficient methods for finding the product MN, where M and N are $n \times n$ matrices and at least one of them is centrosymmetric or skew-centrosymmetric.

We employ the following notation. We denote the transpose of a matrix A by A^T. We use the notation $\lfloor x \rfloor$ for the largest integer less than or equal to x. As usual, I denotes the identity matrix. Throughout this paper we let $\delta = \lfloor \frac{n}{2} \rfloor$. By the *main counterdiagonal* (or simply *counterdiagonal*) of a square matrix we mean the positions which proceed diagonally from the last entry in the first row to the first entry in the last row. We mean by the time complexity the number of flops. When counting flops, we treat addition/subtraction the same as multiplication/division.

Definition 1.1. The *counteridentity* matrix, denoted J, is the square matrix whose elements are all equal to zero except those on the counterdiagonal, which are all equal to 1.

Multiplying a matrix A by J from the left results in reversing the rows of A and multiplying A by J from the right results in reversing the columns of A. Throughout this paper J is used to denote the counteridentity matrix.

There are various kinds of symmetries that we will use in this paper. For convenience, we summarize them in the following definition.

Definition 1.2. Let A be an $n \times n$ matrix.

(1) A is *persymmetric* if $JAJ = A^T$.
(2) A is *centrosymmetric* if $JAJ = A$.
(3) A is *skew-centrosymmetric* if $JAJ = -A$.

Centrosymmetric and skew-centrosymmetric matrices arise in many fields including communication theory, statistics, physics, harmonic differential quadrature, differential equations, numerical analysis, engineering, sinc methods, magic squares, and pattern recognition. For applications of these matrices, see [1, 3, 4, 5, 6, 8]. Note that symmetric Toeplitz matrices are symmetric centrosymmetric, and skew-symmetric Toeplitz matrices are skew-symmetric skew-centrosymmetric.

Now we state the following two theorems that can be proved easily. The second theorem can be found in [2].

<u>Theorem 1.3</u>. Let S be an $n \times n$ skew-centrosymmetric matrix. If n is even, then S can be written as

$$S = \begin{bmatrix} A & -JCJ \\ C & -JAJ \end{bmatrix},$$

where A, J and C are $\delta \times \delta$. If, in addition, S is skew-symmetric, then A is skew-symmetric and C is persymmetric. If n is odd, then S can be written as

$$S = \begin{bmatrix} A & z & -JCJ \\ y & 0 & -yJ \\ C & -Jz & -JAJ \end{bmatrix},$$

where A, J, and C are $\delta \times \delta$, z is $\delta \times 1$, and y is $1 \times \delta$. If, in addition, S is skew-symmetric, then $y = -z^T$, A is skew-symmetric, and C is persymmetric.

<u>Theorem 1.4</u>. Let H be an $n \times n$ centrosymmetric matrix. If n is even, then H can be written as

$$H = \begin{bmatrix} A & JCJ \\ C & JAJ \end{bmatrix},$$

where A, J, and C are $\delta \times \delta$. If, in addition, H is symmetric, then A is symmetric and C is persymmetric. If n is odd, then H can be written as

$$\begin{bmatrix} A & z & JCJ \\ y^T & q & y^T J \\ C & Jz & JAJ \end{bmatrix},$$

where A, J, and C are $\delta \times \delta$, z and y are $\delta \times 1$, and q is a scalar. If, in addition, H is symmetric, then $y = z$, A is symmetric, and C is persymmetric.

2. Algorithms. We present simple efficient algorithms for solving $Gx = b$, where G is centrosymmetric/skew-centrosymmetric and x and b are vectors. We do that by transforming the problem to solving two linear systems "half" the size of the original one. First, we handle centrosymmetric matrices of even order, then centrosymmetric matrices of odd order, then skew-centrosymmetric matrices, then other matrices such as Hermitian persymmetric matrices.

2.1 Centrosymmetric Matrices of Even Order. Let H be an $n \times n$ centrosymmetric matrix, where n is even, let x and b be $n \times 1$ vectors, let H be decomposed as in Theorem 1.4, and let

$$Q_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ -J & J \end{bmatrix},$$

where I and J are $\delta \times \delta$. Then, ${Q_1}^T H Q_1 = D_1$, where

$$D_1 = \begin{bmatrix} L & 0 \\ 0 & M_1 \end{bmatrix},$$

where $L = A - JC$ and $M_1 = A + JC$. Now let

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ and } b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

where x_1, x_2, b_1, b_2 are $\delta \times 1$. Thus, $Hx = b$ if and only if $D_1 {Q_1}^T x = {Q_1}^T b$ if and only if

$$\begin{bmatrix} L & 0 \\ 0 & M_1 \end{bmatrix} \begin{bmatrix} x_1 - Jx_2 \\ x_1 + Jx_2 \end{bmatrix} = \begin{bmatrix} b_1 - Jb_2 \\ b_1 + Jb_2 \end{bmatrix}.$$

The last equation can be simplified to

$$\begin{bmatrix} L{x_1}' \\ M_1{x_2}' \end{bmatrix} = \begin{bmatrix} {b_1}' \\ {b_2}' \end{bmatrix},$$

where ${x_1}' = x_1 - Jx_2$, ${x_2}' = x_1 + Jx_2$, ${b_1}' = b_1 - Jb_2$, and ${b_2}' = b_1 + Jb_2$.

Thus, to solve $Hx = b$ for x:

(1) Find L, M_1, ${b_1}'$, and ${b_2}'$.
(2) Solve the two systems $L{x_1}' = {b_1}'$ and $M_1{x_2}' = {b_2}'$ for x_1' and x_2'.
(3) Obtain the solution

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

of the original system from ${x_1}'$ and ${x_2}'$ as follows:

$$x_1 = \frac{1}{2}\left({x_1}' + {x_2}'\right), \quad x_2 = \frac{1}{2}J\left({x_2}' - {x_1}'\right).$$

Note that the time complexity of the first step of the algorithm is $\frac{1}{2}n^2 + O(n)$ ($\frac{1}{2}n^2 + n$ additions/subtractions and no multiplications/divisions) and the time complexity of the third step is $O(n)$ (n additions/subtractions and n multiplications). The time complexity of the second step depends on the method used to solve the systems. The second step is the step that leads to the reduction of the time complexity, because instead of solving a linear system of n equations, we end up solving two linear systems half the size of the original one. For example, if the original system is solved with Gaussian elimination, then the time complexity will be $\frac{2}{3}n^3 + O(n^2)$. But, if Gaussian elimination is used to solve the two systems in the second step, then the time complexity of our algorithm will be $\frac{1}{6}n^3 + O(n^2)$, which is a significant reduction. If a method more efficient than Gaussian elimination is used, then the time complexity of our algorithm will be less. If, in addition, H is symmetric or skew-symmetric or Toeplitz, then the time complexity of our algorithm will reduce further.

2.2 Centrosymmetric Matrices of Odd Order. Let H be an $n \times n$ centrosymmetric matrix, where n is odd, let x and b be $n \times 1$ vectors, let H be decomposed as in Theorem 1.4, and let

$$Q_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} I & 0 & I \\ 0 & \sqrt{2} & 0 \\ -J & 0 & J \end{bmatrix},$$

where I and J are $\delta \times \delta$. Then, ${Q_2}^T H Q_2 = D_2$, where

$$D_2 = \begin{bmatrix} L & 0 \\ 0 & M_2 \end{bmatrix},$$

where $L = A - JC$ and

$$M_2 = \begin{bmatrix} q & \sqrt{2}y^T \\ \sqrt{2}z & A + JC \end{bmatrix}.$$

Now let

$$x = \begin{bmatrix} x_1 \\ \alpha \\ x_2 \end{bmatrix} \text{ and } b = \begin{bmatrix} b_1 \\ \beta \\ b_2 \end{bmatrix},$$

where x_1, x_2, b_1, b_2 are $\delta \times 1$ and α and β are numbers. Then $Hx = b$ if and only if $D_2 {Q_2}^T x = {Q_2}^T b$ if and only if

$$\begin{bmatrix} L{x_1}' \\ M_2 x_3 \end{bmatrix} = \begin{bmatrix} {b_1}' \\ b_3 \end{bmatrix},$$

where ${x_1}' = x_1 - Jx_2$, ${b_1}' = b_1 - Jb_2$,

$$x_3 = \begin{bmatrix} \sqrt{2}\alpha \\ {x_2}' \end{bmatrix},\ b_3 = \begin{bmatrix} \sqrt{2}\beta \\ b_1 + Jb_2 \end{bmatrix},$$

and ${x_2}' = x_1 + Jx_2$.

Thus, to solve $Hx = b$ for x:

(1) Find L, M_2, ${b_1}'$, and b_3.
(2) Solve the two systems $L{x_1}' = {b_1}'$ and $M_2 x_3 = b_3$ for x_1' and x_3.
(3) Let

$$x_3 = \begin{bmatrix} \gamma \\ {x_2}' \end{bmatrix},$$

where ${x_2}'$ is $\delta \times 1$. Obtain the solution

$$x = \begin{bmatrix} x_1 \\ \alpha \\ x_2 \end{bmatrix}$$

of the original system from $x_1{}'$ and x_3 as follows:

$$x_1 = \frac{1}{2}\left(x_1{}' + x_2{}'\right), \; \alpha = \frac{1}{\sqrt{2}}\gamma, \; x_2 = \frac{1}{2}J\left(x_2{}' - x_1{}'\right).$$

Note that the time complexity of the first step of the algorithm is $\frac{1}{2}n^2 + O(n)$ and the time complexity of the third step is $O(n)$. The time complexity of the second step depends on the method used to solve the systems. (See the last paragraph of the last subsection.)

2.3 Skew-centrosymmetric Matrices. The algorithm for skew-centrosymmetric matrices of even order can be derived in a similar way to the algorithm for centrosymmetric matrices of even order, but here we use the decomposition described in Theorem 1.3. Another way to do it is as follows. Let S be an $n \times n$ skew-centrosymmetric matrix, where n is even, let b be an $n \times 1$ vector, let $m = \frac{n}{2}$, and let

$$E = \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix},$$

where I is the $m \times m$ identity matrix. It is easy to see $H = ES$ is centrosymmetric. Thus, to solve $Sx = b$ for x, solve $Hx = Eb$ by the algorithm of Subsection 2.1.

2.4 Complex Systems. Let $R = H + iS$, where H is an $n \times n$ real centrosymmetric matrix and S is an $n \times n$ real skew-centrosymmetric matrix. Let $X = X_1 + iX_2$ and $B = B_1 + iB_2$, where X_1, X_2, B_1, and B_2, are $n \times 1$ real vectors. Then, $RX = B$ if and only if

$$\begin{bmatrix} H & -S \\ S & H \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}.$$

Note that the above system can be written as

$$\begin{bmatrix} H & JSJ \\ S & JHJ \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}.$$

Now we can use the algorithm of Subsection 2.1 to solve this system, which leads to a reduction in the time complexity. Thus, instead of solving the complex system $RX = B$, we can solve the two real systems $(H - JS)X_1' = B_1'$ and $(H + JS)X_2' = B_2'$, where $B_1' = B_1 - JB_2$ and $B_2' = B_1 + JB_2$. The solution of the complex system is $\frac{1}{2}(X_1' + X_2') + \frac{1}{2}iJ(X_2' - X_1')$. For example, if R is an $n \times n$ Hermitian persymmetric matrix, then $R = H + iS$, where H is an $n \times n$ real symmetric centrosymmetric matrix and S is an $n \times n$ real skew-symmetric skew-centrosymmetric matrix. Hence, we can use the idea described above to solve the complex system $RX = B$.

3. Multiplication of Matrices. Let M and N be $n \times n$ matrices and let at least one of them be centrosymmetric/skew-centrosymmetric. To reduce the time complexity of finding the product MN, we use similar ideas to those we used in the previous section. For example, if M is centrosymmetric of even order, then we replace M by $Q_1 D_1 Q_1^T$ (see Subsection 2.1). If the standard multiplication algorithm is used, then the reduction in time complexity that results from using this idea is as follows:

(1) About 75% if both matrices are centrosymmetric/skew-centrosymmetric (or one is centrosymmetric and the other is skew-centrosymmetric). Here, instead of multiplying two $n \times n$ matrices, we end up multiplying two matrices "half" the size.

(2) About 50% if one of the matrices is arbitrary and the other is centrosymmetric/skew-centrosymmetric. Here, instead of multiplying two $n \times n$ matrices, we end up multiplying four matrices "half" the size.

Note that a more efficient algorithm like Strassen's algorithm or Coppersmith and Winograd's algorithm can be used to multiply the matrices "half" the size mentioned above instead of using the standard multiplication algorithm.

References

1. A. L. Andrew, "Centrosymmetric Matrices," *SIAM Rev.,* 40 (1998), 697–698.

2. A. Cantoni and P. Butler, "Eigenvalues and Eigenvectors of Symmetric Centrosymmetric Matrices," *Linear Algebra Appl.,* 13 (1976), 275–288.

3. A. Cantoni and P. Butler, "Properties of the Eigenvectors of Persymmetric Matrices With Applications to Communication Theory," *IEEE Trans. Comm.,* **COM-24** No. 8 (1976), 804–809.

4. L. Datta and S. Morgera, "On the Reducibility of Centrosymmetric Matrices - Applications in Engineering Problems," *Circuits Systems Signal Process,* 8 (1989), 71–96.

5. I. J. Good, "The Inverse of a Centrosymmetric Matrix," *Technometrics,* 12 (1970), 925–928.

6. R. B. Mattingly, “Even Order Regular Magic Squares Are Singular,” *Amer. Math. Monthly,* 107 (2000), 777–782.

7. A. Melman, “Symmetric Centrosymmetric Matrix-Vector Multiplication,” *Linear Algebra Appl.,* 320 (2000), 193–198.

8. F. Stenger, “Matrices of Sinc Methods,” *J. Comput. Appl. Math.,* 86 (1997), 297–310.

Mathematics Subject Classification (2000): 65F05

Iyad T. Abu-Jeib
Department of Computer Science and Information Systems
SUNY Fredonia
Fredonia, NY 14063
email: abu-jeib@cs.fredonia.edu