Indian J. pure appl. Math., 19 (6) : 554-561, June 1988

ON SYMMETRIZING A MATRIX

S. K. SEN AND V. CH. VENKAIAH*
Super computer Education and Research Laboratory, Indian Institute of Science, Bangalore 560012

(*Received* 13 *October* 1986; *after revision* 29 *February* 1988)

A symmetrizer of an abritrary matrix A is a matrix X satisfying the matrix equations $XA = A^tX$ and $X = X^t$. A symmetrizer is useful in transforming an apparently nonsymmetric eigenvalue problem into a symmetric one. An algorithm to compute a matrix symmetrizer is presented and illustrated.

1. INTRODUCTION

A symmetrizer of an arbitrary square matrix A is a symmetric solution X that satisfies $XA = A^tX$, where t indicates the transpose. Symmetrizers reduce a nonsymmetric (real) eigenvalue problem (nep) into a symmetric eigenvalue problem (sep) which is relatively easy to solve and are useful in the stability problems of control theory[1]. These are also useful in the study of general matrices[9].

Transformation of nep to sep using a symmetrizer—Let A be a square nonsymmetric real matrix and X be a nonsingular symmetrizer of A. Then the nep $Ay = \lambda y$ can be reduced to an sep $Bx = \lambda x$ as follows. $Ay = \lambda y$ implies $XAy = \lambda Xy$. Since X is real symmetric, we have $X = P^t DP$, where P is orthogonal (i.e., $P^t = P^{-1}$) and $D = (d_{ii})$ is diagonal. Therefore, we have $PXAy = \lambda DPy$. Let $D_1 = (d^{\frac{1}{2}}_{ii})$ be a diagonal matrix such that $D_1^2 = D$. Then $PXAy = \lambda D_1^2\, Py$. $d_{ii} \neq 0$ for all i since X has nonzero eigenvalues (as X is nonsingular). Hence D_1^{-1} exists. Thus, $Bx = \lambda x$, where the symmetric matrix, real or complex,

$$B = D_1^{-1}\, PX\, AP^t\, D_1^{-1} \qquad \ldots(1)$$

and $x = D_1\, Py$. If X is positive definite then $d_{ii} > 0$ and consequently $d^{\frac{1}{2}}_{ii}$ is real. In this case B is real symmetric. If X has a negative eigenvalue then B is complex symmetric.

Symmetrizer to compute zeros of a real polynomial—A symmetrizer can be used to obtain the roots of a polynomial equation[5]. The associated companion matrix[8] of the polynomial

* supported by CSIR.

$$p_n(x) = \sum_{i=0}^{n} a_i x^i$$

where $a_n = 1$, and a_i $i = 0, 1, \ldots, n-1$ are real, is

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & -a_0 \\ 1 & 0 & \ldots & 0 & -a_1 \\ \vdots & & & & \vdots \\ 0 & 0 & \ldots & 1 & -a_{n-1} \end{bmatrix}$$

whose eigenvalues are the zeros of $p_n(x)$. To obtain these eigenvalues, we compute an X such that $XA = A^t X$ and $X = X^t$ and then we compute the eigenvalues of the symmetric matrix $B = D_1^{-1} \; PXAP^t \; D_1^{-1}$, by a method, say, the Jacobi method[8,11] if the matrix X is positive definite. If the nonsingular X has one or more negative eigenvalues then a Cholesky-type decomposition[8,11] can be used to obtain the eigenvalues of B.

Existence of a symmetrizer—There exists a nonsingular symmetrizer for any square matrix[10]. Let A be a real nonsymmetric matrix and X be one of its symmetrizers. Then the following table depicts the scope of the symmetrizer X.

TABLE I

Scope of a symmetrizer

Eigenvalues of X	*Reduction of nep to sep*	*Utility*
All positive (X is positive definite)	Possible	Useful since sep is easier to solve (say, by Jacobi method)
One or more zero (X is singular)	Not possible	—
One or more negative and no zero	Possibe	Complex arithmetic needs to be used to solve sep (say, by Cholesky-type decomposition).

Since X is not unique, one can always compute another X by a different choice of some of the elements of X (see sec. 2) so that X is positive definite, a positive definite X does not, however, exist if the matrix A has complex eigenvalues.

2. Algorithm to Compute a Matrix Symmetrizer

Let $A = (a_{ij})$ be an $n \times n$ real nonsymmetric matrix. Also, let $X = (x_{ij})$ be a symmetrizer of A to be computed.

Step 1 : [Compute A_{ii}]

For $i := 1$ to $n - 1$ do

compute the $(n-i)\times(n-i+1)$ matrix A_{ii},

where (k, t) th element of A_{ii}

$:= a_{t+i-1,k+i} - a_{ii}$ if $t+i-1 = k+i$ else

$:= a_{t+i-1,k+i}$.

Step 2 : [Compute A_{ij} $j > i$]

For $i := 1$ to $n-1$ do

compute the $(n-i)\times(n-j+1)$ matrix A_{ij}

where (k, t) th element of A_{ij}

$:= 0$ for $k := 1$ to $j-i-1$

$:= 0$ for $k := j-i+1$ to $n-i$ and $t \neq k-j+i+1$

$:= -a_{j+t-1,t}$ for $k := j-i$

$:= -a_{ji}$ for $k := j-i+1$ to $n-i$ and $t = k-j+i+1$.

Step 3 : [Compute A_{ij} $j < i$]

For $i := 2$ to $n-1$ do

compute the $(n-i)\times(n-j+1)$ matrix A_{ij},

where (k, t) th elememet of A_{ij}

$:= 0$ for $t := 1$ to $i-j$

$:= 0$ for $t := i-j+2$ to $n-j+1$ and $k \neq t+j-i-1$

$:= a_{j,i+k}$ for $t := i-j+1$

$:= -a_{ji}$ for $t := i-j+2$ to $n-j+1$ and $k = t+j-i-1$.

Step 4 : [Obtain C]

Obtain $C = (A_{ij})$ $i = 1, 2, \ldots, n-1, j = 1, 2, \ldots, n$.

Remark : C is an $(n^2-n)/2 \times (n^2+n)/2$ matrix.

Step 5 : [Solve $Cz = 0$]

Solve $Cz = 0$,

where

$$z = [x_{11}\ x_{12} \ldots x_{1n}\ x_{22}\ x_{23} \ldots x_{2n} \ldots x_{n-1,n-1}\ x_{n-1,n}\ x_{nn}]$$

as follows.

Step 5a : Choose, if none of the columns of C is zero, n elements $x_{11} = c \neq 0$, $x_{12} = x_{13} = \ldots = x_{1n} = 0$ or any n elements x_{ij} $(i \leqq j)$ so that the resulting equations are consistent. If a column, which constitutes the coefficients of some x_{ij}, of C is zero then choose x_{ij} along with any other $n - 1$ elements of X so that the system is consistent and at least one of these $n - 1$ elements is nonzero to ensure nonsingularity of X. If k columns of C are zero then choose the corresponding x_{ij} along with any other $n - k$ elements of X (without disturbing consistency) so that at least one of these $n - k$ elements is nonzero to ensure nonsingularity of X. Let the resulting equations be $C' z' = b'$.

Step 5b : Use any method, say, the Gauss elimination method[8,11] to solve the equation — this gives X and terminate.

Proof of the algorithm—The proof steps 1 — 4 follows from the expansion and rearrangement of the equations $XA = A^t X$, $X^t = X$ and that of Step 5 (solving linear equations) is well-known.

2.1. *Stability of the algorithm*—The algorithm comprises (a) computing C, which is always stable, and (b) solving the linear equations $C' z' = b'$. Therefore, the numerical stability of the algorithm essentially depends on the degree of singularity of the matrix C' and on the stability of the method that is employed to solve $C' z' = b'$. The choice of the elements in Step 5 decides whether C' is near-singular, i.e., ill-conditioned with respect to the inverse, or not. There is, however, no easy way to know beforehand which will be a good choice. The selection of a stable method to solve linear equations usually poses no problem.

If the matrix whose symmetrizer $X = (x_{ij})$, where $x_1, x_2, \ldots, x_n$ are the rows of X, is required is lower Hessenberg and if we choose x_n of the form $[c\ 0\ 0\ \ldots\ 0]$, where $c \neq 0$ (see Sec. 4 also) then the algorithm is stable provided the Hessenberg matrix has the codiagonal nonzero. If any one element of the codiagonal is near-zero then the algorithm tends to become unstable unless some other appropriate choice is made.

3. Example

We compute a symmetrizer of the matrix $A = (a_{ij})$, where

$$A = \begin{bmatrix} 5 & 1 & 0 \\ 0 & 3 & 0 \\ 1 & -1 & 2 \end{bmatrix}.$$

Here

$$C = \left[\begin{array}{ccc|cc|c} 1 & -2 & -1 & 0 & -1 & 0 \\ 0 & 0 & -3 & 0 & 0 & -1 \\ \hline 0 & 0 & -1 & 0 & -1 & 1 \end{array}\right]$$

and

$$z = [x_{11}\ x_{12}\ x_{13}\ x_{22}\ x_{23}\ x_{33}]^t.$$

Choice 1—$x_{22} = 3$ (this choice is made since the corresponding column is null), $x_{11} = 2$, and $x_{12} = 1/2$. Any other choice for any three elements of X may be made preserving the consistency of the equations. Choice 1 gives the nonhomogeneous equations $C'\ z' = b'$, where

$$C' = \begin{bmatrix} -1 & -1 & 0 \\ -3 & 0 & -1 \\ -1 & -1 & 1 \end{bmatrix}, z' = \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \end{bmatrix}, b' = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

Applying the Gauss reduction method with partial pivoting and then using the back substitution, we obtain $x_{33} = 1$, $x_{23} = 4/3$, $x_{13} = -1/3$. Hence the symmetrizer is

$$X = \begin{bmatrix} 2 & 1/2 & -1/3 \\ 1/2 & 3 & 4/3 \\ -1/3 & 4/3 & 1 \end{bmatrix}$$

which is positive definite. The symmetrizer transforms nep $Ay = \lambda y$ to sep $Bx = \lambda x$, where $B = D_1^{-1}\ PXAP^t\ D_1^{-1}$, $X = P^t\ DP$, and $D = D_1^2$ for some diagonal matrix D_1. From X, we compute using the Jacobi method

$$P = \begin{bmatrix} .1787430 & .8919905 & .4152155 \\ -.9434132 & .0355713 & .3297064 \\ .2793252 & -.4506525 & .8478737 \end{bmatrix},$$

$$D = \begin{bmatrix} 3.720851 & 0 & 0 \\ 0 & 2.097642 & 0 \\ 0 & 0 & 0.1815073 \end{bmatrix}.$$

Taking the positive square-root of the diagonal elements of D, we obtain D_1 (negative square-root can also be taken). Hence

$$B = \begin{bmatrix} 2.7547790 & -1.1969110 & -0.1342376 \\ -1.1969110 & 4.3387318 & -0.4728604 \\ -0.1342376 & -0.4728604 & 2.9302057 \end{bmatrix}.$$

It can be easily checked that the eigenvalues (computed by the Jacobi method) of B, viz., 5.042097, 2.999203, 1.982418 are approximately the same as those of the real nonsymmetric matrix A, which are 5, 3, 2.

Choice 2—$x_{22} = 1$, $x_{11} = 1$, $x_{12} = 0$. Solving the resulting nonhomogeneous equations, we obtain the nonsingular symmetrizer

$$X = \begin{bmatrix} 1 & 0 & -1/3 \\ 0 & 1 & 4/3 \\ -1/3 & 4/3 & 1 \end{bmatrix}.$$

having a negative eigenvalue, viz., -0.3743685. Hence $B = D_1^{-1}\,PXAP'\,D_1^{-1}$

$$= \begin{bmatrix} 1.8348370 & -0.2159493 & i0.4159453 \\ -0.2159492 & 5.1176470 & i0.5438464 \\ i0.4159453 & i0.5438464 & 3.0475158 \end{bmatrix},$$

where

$$D = \begin{bmatrix} 2.3743690 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -0.3743685 \end{bmatrix}$$

and

$$P = \begin{bmatrix} 0.1714986 & -0.6859943 & -0.7071068 \\ 0.9701425 & 0.2425356 & -0.7304784E\text{-}20 \\ 0.1714986 & -0.6859943 & 0.7071068 \end{bmatrix}.$$

The eignvalues of the complex symmetric matrix B can be computed by a Cholesky-type decomposition; however, it can be easily checked that the eigenvalues of B are approximately the same as those of A, viz., 2, 3, 5.

All the foregoing computations have been carried out on DEC 1090-TOPS 10 computing system in single precision and 7 decimal digits are input.

4. Recursive Algorithm for a Lower Hessenberg Matrix

A simple and elegant recursive algorithm to compute a symmetrizer was suggested by Datta[2]. The algorithm is as follows. Let $A = (a_{ij})$ be an $n \times n$ lower Hessenberg matrix with nonzero codiagonal. Let $x_1, x_2, \ldots, x_n$ be the rows of a symmetrizer X to be computed.

Step 1 : Choose $x_n \neq 0$ arbitrarily. (To ensure nonsingularity of X, choose x_n of the form $[c\ 0\ 0 \ldots 0]$ where $c \neq 0$.)

Step 2 : Compute $x_{n-1}, x_{n-2}, \ldots x_1$ recursively from

$$x_i = (1/a_{i,i+1})\,(x_{i+1}A - a_{i+1,i+1}\,x_{i+1} - a_{i+2,i+1}\,x_{i+2} - \ldots - a_{n,i+1}\,x_n).$$

This algorithm is applicable only to lower Hessenberg matrices with nonzero codiagonal elements since each codiagonal element $a_{i,i+1}$ occurs in the denominator of Step 2 of the algorithm. However, any matrix can be reduced to a lower Hessenberg form by using similarity transformations involving Gauss-type elimination and this reduction is direct; thus, the eigenvalues remain invariant.

Derivation of the recursive algorithm—The foregoing recursive algorithm is a special case of the suggested general algorithm and is derived as follows. Consider the homogeneous equations $Cz = 0$ for the $n \times n$ nonsymetric real matrix $A = (a_{ij})$, where C and z are defined in Sec. 2. Substitute $a_{ij} = 0$ for $j = i + 2, i + 3, \ldots, n$ in C since A is assumed to be lower Hessenberg. Choose (as suggested by Datta) $x_{n1} = x_{1n} = c \neq 0, x_{n2} = x_{2n} = 0, \ldots, x_{nn} = 0$ in z and write the resulting linear equations $C' z' = b'$, where C' is a triangular matrix. Using back substitution, the recursive algorithm follows.

If A has any other special property then this can be incorporated in the general algorithm so that the general algorithm reduces to a simpler (special) one.

5. Conclusions

A general algorithm for computing a symmetrizer of an arbitray matrix A is suggested. A case where A is lower Hessenberg is shown to be a special case of the general algorithm. If the nonsymmetric matrix has any other special structure then this structure can be exploited to generate a special case of the algorithm. The advantage of obtaining a symmetrizer for computing the eigenvalues of a nonsymmetrix is yet to be fully investigated.

The computation of a symmetrizer involves a finite number of arithmetic operations and hence a symmetrizer can be computed exactly using finite-field transform techniques[3,4,6,7] superimposed on the (direct) method of solving linear equations. The need to have an exact symmetrizer arises when errors involved due to the use of real arithmetic is considerable.

Acknowledgement

The authors wish to express sincere thanks to the referees for constructive comments.

References

1. S. Barnett, Some topics in algebraic systems theory : a survey, *Recent Mathematical Developments in Control*, (ed. : D. J. Bell). Academic Press, New York, 1973, p. 323.
2. B. N. Datta, An algorithm for computing a symmetrizer of a Hessenberg matrix (unpublished).
3. R. T. Gregory, *Tech. Rep. CS-84-57, Computer Science Department, Univ. of Tennessee, Knoxville, TN 37916*, July 1984.
4. R. T. Gregory, and E. V. Krishnamurthy, *Methods and Applications of Error-free Computations*, Springer-Verlag; New York, 1984.
5. J. L. Howland, *Am. Math. Mon.* 68 (1961), 235.
6, E. V. Krishnamurthy, T. M. Rao, and K. Subramanian, *Proc. Indian Acad. Sci* 81 A (1975), 58.
7. E. V. Krishnamurthy, T. M. Rao, and K. Subramanian, *Proc. Indian Acad. Sci.* 82 A (1975), 165.

8. E. V. Krishnamurthy, and S. K. Sen, *Numerical Algorithms : Computations in Science and Engineering*, Affiliated East-West Press, New Delhi, 1986.
9. O. Taussky, *Linear Algebra and Appl.* 5 (1972), 147.
10. O. Tausaky, and H. Zassenhaus, *Pacific J. Math.* 9 (1959), 893.
11. J. H. Wilkinson, *Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.