Mathematics Notes

Note 24

April 1972

CDCSM and SOL

Subroutines Which Return the Solution of Simultaneous Linear Equations When the System is Complex and the Matrix Formed by the System is Symmetric

Joe P. Martinez

The Dikewood Corporation

ABSTRACT

This note describes computer subroutines which are based on the Choleski method for the solution of a system of simultaneous linear equations of the form $A\vec{x} = \vec{b}$, where A is symmetric, by triangular decomposition.

I. INTRODUCTION

In certain classes of EMP investigations the problem is formulated into a system of simultaneous linear equations. The solution is then obtained by some matrix inversion technique. In many cases the matrix which is formed is symmetric about the main diagonal. One may take advantage of the symmetry properties in the matrix by utilizing special methods in order to increase the speed of the solution calculations. This note describes one such method.

II. OPERATION

General

The subroutines CDCSM (Choleski's Decomposition of a Complex Symmetric Matrix) and SOL (Solution) are called by a main program or subprogram which fill the matrix array with the coefficients of the system of equations to be solved. CDCSM will return to the calling program the decomposed matrix. SOL is then called by the calling program with the vector $\vec{b}$ in

$$A\vec{x} = \vec{b} \tag{1}$$

and the solution vector $\vec{x}$ is returned.

Many different $\vec{b}$ vectors may be used once A has been decomposed. Since A is symmetric, only the lower triangular half plus the diagonal need to be filled in the array containing the elements of A. The arrays which contain A, $\vec{x}$, and $\vec{b}$ must be typed COMPLEX and dimensioned in the calling routine.

Calling Sequence

To use subroutines CDCSM and SOL, the calling routine must supply the standard FORTRAN IV CALL statements,

```
CALL CDCSM (A, L, N, DETER)
```

and

```
CALL SOL (A, L, N, B)
```

Parameters

The following are parameters for CDCSM

1.	A	TYPE COMPLEX. This is a two-dimensional complex array containing the elements of the matrix. The first dimension relates to the row position and the second to the column position. Thus the matrix element A_{ij} is contained in A(I, J). CDCSM returns the decomposed matrix in A.
2.	L	TYPE INTEGER. This integer is supplied to the subroutine and must be the same number as that used in the dimension statement where A is dimensioned in the calling routine. That is, if the dimension statement is DIMENSION A(45, 45) then L must be 45.
3.	N	TYPE INTEGER. The order of the system to be solved. $N \leq L$.
4.	DETER	TYPE COMPLEX. The determinant of the system, returned by CDCSM.

The following are parameters for SOL

1.	A	TYPE COMPLEX. Same as for CDCSM. This is an input for SOL and is the decomposed matrix returned by CDCSM.
2.	L	TYPE INTEGER. Same as for CDCSM.
3.	N	TYPE INTEGER. Same as for CDCSM.
4.	B	TYPE COMPLEX. An array dimensioned N, this is the $\vec{b}$ vector in $A\vec{x} = \vec{b}$. The solution vector $\vec{x}$ is returned in B by SOL.

Names given the above parameters were chosen for illustrative purposes. These are formal parameters in the subroutines. The actual parameters used by the calling routine need not agree in name, but they must agree in type and number.

III. GENERAL

Accuracy

These subroutines were tested on the Control Data Corporation 6600 at Kirtland Air Force Base, New Mexico, which can carry floating point numbers to 14 or 15 significant places. Symmetric matrices were generated with random numbers for the complex elements. These random numbers ranged between -100 and +100. The $\vec{b}$ vector was selected such that the $\vec{x}$ would have 1 in each element, that is,

$$\begin{bmatrix} a_{11} a_{12} \cdots a_{1n} \\ a_{21} a_{22} \cdots & \\ \vdots \quad \vdots & \\ & \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \sum_j a_{1j} \\ \sum_j a_{2j} \\ \vdots \\ \sum_j a_{nj} \end{bmatrix} \tag{2}$$

This provides a quick check for accuracy since the $\vec{x}$ vector should contain all ones. A root mean square difference of the form

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (1-x_i)^2}{n}} \tag{3}$$

was used to see what errors the routines generated. For a 5 x 5 matrix, σ was in the order of 2×10^{-14}. For a 50 x 50, σ was $\sim 2 \times 10^{-13}$ and for a 150 x 150, $\sigma \sim 1 \times 10^{-12}$. As with any matrix routines, much depends on the conditioning of the matrix. Since the properties of a symmetric matrix are being used one cannot employ pivoting to reduce error since that would destroy symmetry properties. Many references exist[1, 2] which discuss in detail the error analysis of such calculations.

Time

CDCSM decomposed random symmetric matrices of various sizes from 5 x 5 to 150 x 150. For a 5 x 5 it took roughly .003 seconds; for a 50 x 50 about .5 seconds, and for a 150 x 150 about 11.3 seconds. So, if a matrix is of size n x n a very rough estimate of the time it will take to decompose is $.0004n^2$ seconds.

This is not an exact figure and serves only as a guideline. Of course the speed of the calculations varies greatly with different computers. The SOL routine returned an answer vector with $\sim .00001n^2$ seconds elapsed time.

Storage

Subroutine CDCSM takes 251_8 words of storage and SOL takes 175_8 on the CDC 6600.

Checks

There are no checks in the subroutines. The user should be familiar enough with the system of equations to ensure non-singularity. In addition, inaccuracies may result because of an ill-conditioned matrix, or if the matrix or its elements are large in absolute value, roundoff error may result. Also, values beyond machine limits ($\pm 10^{-294}$ to $\pm 10^{322}$ in the CDC 6600) may be generated. These problems are the same with any matrix inversion scheme, but may be worse with this one since pivoting cannot be used.

IV. ALGORITHM

Choleski Scheme

The algorithm is that of Choleski and is a variation of Gaussian elimination. It is also called the square root method.

In regular triangular decomposition the matrix A is decomposed into an upper triangular and lower triangular matrix U and L such that the elements of the diagonal in L are all ones, and the product of U and L is A, that is,

$$A = LU \tag{4}$$

In L, the elements above the diagonal are all zero, and similarly in U the elements below the diagonal are zero.

The solution is then found by solving the system

$$L\vec{y} = \vec{b} \tag{5}$$

and then

$$U\vec{x} = \vec{y} \tag{6}$$

since

$$A\vec{x} = LU\vec{x} = L\vec{y} = \vec{b} \tag{7}$$

Now, when A is symmetric it may be written as,

$$A = LL^T \tag{8}$$

where L^T is the transpose of L, if the decomposition is modified to produce two triangular matrices in which the diagonal elements of L and U are the same. The algorithm as given in Ref. 1 and 2 is

$$\ell_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2} \qquad j = 1, \ldots, n \tag{9}$$

$$\ell_{ij} = \frac{1}{\ell_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk}\right) \qquad \begin{array}{l} i = j+1, \ldots, n \\ j = 1, \ldots, n \end{array} \tag{10}$$

Note that the elements of A above the diagonal are not involved due to the symmetry.

The determinant of the matrix is found by multiplying the diagonal elements of L with each other then squaring the results. This is derived from

$$\mathrm{DET}(A) = \mathrm{DET}(L)\mathrm{DET}(U) = \mathrm{DET}(L)\mathrm{DET}(L^T) = \mathrm{DET}^2(L) \tag{11}$$

Since L is a lower triangular matrix, its determinant is simply the product of the diagonal elements. L^T is an upper triangular matrix with the same diagonal elements. So, the determinant of A will be the square of the product of the diagonal elements of the decomposed matrix. Since no row interchanges occur, we need not keep track of sign changes in the determinant.

In CDCSM the array A is decomposed by formulas 9 and 10 and the L matrix is stored in A, so, the original matrix is destroyed. L^T is not stored anywhere since in the solution subroutine all one has to do is transpose the indices of L to obtain the elements of L^T.

SOL is called next in order to solve the system of equations. Now we solve for a temporary vector $\vec{y}$. From formula 5, $L\vec{y} = \vec{b}$ we see that

$$\begin{bmatrix} \ell_{11} & 0 & \cdot & \cdot & \cdot & 0 \\ \ell_{21} & \ell_{22} & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \ell_{n1} & \cdot & \cdot & \cdot & \cdot & \ell_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix} \tag{12}$$

so that

$$y_1 = b_1/\ell_{11}$$

$$y_2 = \frac{b_2 - y_1\ell_{21}}{\ell_{22}} = \frac{b_2 - b_1/\ell_{11}}{\ell_{22}} \tag{13}$$

etc.

Next we solve in a similar manner

$$L^T\vec{x} = \vec{y}$$

$$\begin{bmatrix} \ell_{11} & \ell_{21} & \cdot & \cdot & \cdot & \ell_{n1} \\ 0 & \ell_{22} & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \tag{14}$$

so that

$$x_n = y_n/\ell_{nn}$$

$$x_{n-1} = \frac{y_{n-1} - x_n \ell_{n,n-1}}{\ell_{n-1,n-1}} = \frac{y_{n-1} - (y_n/\ell_{nn})\ell_{n,n-1}}{\ell_{n-1,n-1}} \qquad (15)$$

etc.

This is our solution and is returned to the calling program by SOL in the B array.

General

On page 32 of Ref. 1 it is stated that a proof has been made that no general system of linear algebraic equations can be solved in fewer operations than are required by Gaussian elimination. In our decomposition we have further taken advantage of symmetry by deleting the computation of a U matrix. On page 115 of Ref. 1 it further states that because of the symmetry of A it is necessary to store only n(n+1)/2 elements resulting in savings of storage. A relatively complicated subscripting scheme is usually necessary, with consequent loss of time. In this note we did not attempt to save storage. It is possible that the user may do this with the subroutines by adapting them to his particular problem.

It should be noted that REAL rather than COMPLEX variables may be used only if the matrix is positive definite since square roots are involved.

V. SUMMARY

Subroutines CDCSM and SOL may be used to solve a system of simultaneous linear equations if the matrix formed by the system is symmetric, and the coefficients are complex. The algorithm used is a variation of Gaussian Elimination and is called Choleski's method. It should be one of the fastest routines for solving such a system. Samples of accuracy and timing have been given. It was pointed out that care be given to the conditioning of the matrix, since ill-conditioned and singular matrices will return erroneous results.

REFERENCES

1. Forsythe. G., and C. Moler, Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967.

2. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill, Inc., New York, 1965.

3. Perlis, S., Theory of Matrices, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1952.

APPENDIX
PROGRAM LISTING

```
      SUBROUTINE CDCSM(A, L, N, DETER)
C
C     THIS SUBROUTINE DECOMPOSES A COMPLEX SYMMETRIC
C     MATRIX INTO A LOWER TRIANGULAR MATRIX BY CHOLESKI'S
C     METHOD.  IT ALSO CALCULATES THE DETERMINANT OF THE
C     MATRIX.
C
      COMPLEX A(L, N), SUM, DETER
C
      A(1, 1)=CSQRT(A(1, 1))
      DO 1 K=2, N
1     A(K, 1)=A(K, 1)/A(1, 1)
      DETER=A(1, 1)
      DO 4 J=2, N
      SUM=(0., 0.)
      M=J-1
      DO 2 K=1, M
2     SUM=SUM+A(J, K)**2
      A(J, J)=CSQRT(A(J, J)-SUM)
      DETER=DETER*A(J, J)
      IP=J+1
      IF(IP.GT.N)GO TO 5
      DO 4 I=IP, N
      SUM=(0., 0.)
      DO 3 K=1, M
3     SUM=SUM+A(I, K)*A(J, K)
      A(I, J)=(A(I, J)-SUM)/A(J, J)
4     CONTINUE
5     DETER=DETER*DETER
      RETURN
      END
      SUBROUTINE SOL(A, L, N, B)
C
C     THIS SUBROUTINE SOLVES A SYSTEM OF LINEAR EQUATIONS
C     WHEN THE MATRIX IS SYMMETRIC AND COMPLEX.  THE SYS-
C     TEM IS SOLVED BY BACK SUBSTITUTION WITH A LOWER
C     TRIANGULAR MATRIX, WHICH IS RETURNED BY CDCSM AND
C     IS THE DECOMPOSITION OF THE MATRIX FORMED BY THE
C     SYSTEM.
C
      COMPLEX A(L, N), B(N), SUM
C
      DO 2 I=1, N
      B(I)=B(I)/A(I, I)
      IP1=I+1
      IF(IP1.GT.N) GO TO 2
      DO 1 J=IP1, N
1     B(J)=B(J)-A(J, I)*B(I)
2     CONTINUE
      DO 4 K=1, N
      I=N-K+1
      SUM=(0., 0.)
      IP1=I+1
      IF(IP1.GT.N) GO TO 4
      DO 3 J=IP1, N
3     SUM=SUM+A(J, I)*B(J)
4     B(I)=(B(I)-SUM)/A(I, I)
      RETURN
      END
```