Let Δb represent a perturbation in the right-hand side of a linear system.
If $Ax = b$ then

$$A(x+\Delta x) = b+\Delta b$$

where

$$\frac{||\Delta x||}{||x||} \le K(A)\left[\frac{||\Delta b||}{||b||}\right] \qquad (1.1)$$

where $K(A)$ is the condition number of A, $K(A) = ||A||\ ||A^{-1}||$ and $||\cdot||$ is some norm e.g., $||x||_1 = \sum_{i=1}^{n} |x_i|$ if x is a vector.

The methods used in our linear equation package are guaranteed to provide an accurate answer to a slightly perturbed problem. If we assume that our method produces the correct answer to a problem where $||\Delta b|| \le \varepsilon ||b||$, where ε is the machine precision, then on the Honeywell 6000 where ε is about 10^{-8}, a relative error of 4×10^{-5} for the above problem with N of 90 would not be surprising.

```
C
C COMPUTE THE ERROR IN THE SOLUTION
           ERR=0.0
           DO 30 I=1,N
  30           ERR=ERR+ABS(B(I)-1.0)
           ERR=ERR/FLOAT(N)
           IWRITE=I1MACH(2)
           WRITE(IWRITE,31)N
  31       FORMAT(/8H FOR N= ,I5)
           WRITE(IWRITE,32)COND
  32       FORMAT(23H CONDITION ESTIMATE IS 1PE15.7)
           WRITE(IWRITE,33)ERR
  33       FORMAT(30H RELATIVE ERROR IN SOLUTION IS,1PE15.7)
  40     CONTINUE
         STOP
         END
```

The output from the above program run on the Honeywell 6000 at Bell Labs was

```
FOR N=    10
CONDITION ESTIMATE IS   5.4831256E 01
RELATIVE ERROR IN SOLUTION IS  7.3015689E-08

FOR N=    50
CONDITION ESTIMATE IS   1.3551582E 03
RELATIVE ERROR IN SOLUTION IS  4.9673021E-06

FOR N=    90
CONDITION ESTIMATE IS   4.4305656E 03
RELATIVE ERROR IN SOLUTION IS  1.2567225E-05
```

As the output indicates, for small values of N, the matrix is well-conditioned and the solution is very accurate, but as N increases, the matrix becomes more ill-conditioned and the error in the solution increases. Although the relative error for N = 90 appears large, it is not unreasonable as the following analysis indicates:

Method: The Bunch - Kaufman algorithm described in reference [1] below, is used. See reference [2] below for the method used to estimate the condition number.
SYSS calls SYCE and SYFBS.

See also: SYDC, SYFBS, SYMD, SYLE, SYCE

Author: Linda Kaufman

References:
[1] Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

[2] Cline, A. K., Moler, C. B., Stewart, G. W., and Wilkinson, J. H., An estimate for the condition number, *SIAM J. Numer. Anal. 16* (1979), 368-375.

Example: The following program packs the matrix

$$a_{i,j} = | i - j |$$

into the vector C and sets up the right-hand side so that the solution will be all ones. It then calls SYSS to solve the problem and calculates the error in the solution. This example was included to show that seemingly innocent looking problems may be ill-conditioned and to show the effect of ill-conditioning on a solution. It also demonstrates how to pack a symmetric matrix into a vector.

```
         INTEGER N, L, I, IWRITE, I1MACH
         REAL C(5000), B(100)
         REAL SUM, FLOAT, ABS, ERR, COND
         DO 40  N=10,90,40
C
C CREATE THE MATRIX A(I,J)=ABS(I-J), PACK IT INTO
C THE VECTOR C AND FORM THE RIGHT-HAND SIDE SO THE
C SOLUTION HAS ALL ONES.
C
           L=1
           SUM=(N*(N-1))/2
           DO 20 I=1,N
              DO 10 J=I,N
                 C(L)=J-I
                 L=L+1
  10          CONTINUE
              B(I)=SUM
              SUM=SUM+FLOAT(I-(N-I))
  20       CONTINUE
C
C SOLVE THE SYSTEM AND GET THE CONDITION NUMBER OF THE MATRIX
           CALL SYSS(N,C,B,100,1,COND)
```

Note 2: Users who wish to solve a sequence of problems with the same coefficient matrix, but different right-hand sides *not all known in advance*, should not use SYSS, but should call subprograms SYCE and SYFBS. (See the example of SYCE.) SYCE is called once to get the MDM^T decomposition (see the introduction to this chapter) and then SYFBS is called for each new right-hand side.

Error situations: *(The user can elect to 'recover' from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1
2	IB < N
3	NB < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYSS with C, B, and COND declared double precision.

Complex symmetric version: CSYSS with C and B declared complex

Complex Hermitian version: CHESS with C and B declared complex

Storage: N integer locations and
N real (double precision for DSYSS, complex for CSYSS and CHESS) locations of scratch storage in the dynamic storage stack

Time:

$$\frac{N^3}{6} + \left(\frac{19}{4} + NB\right) \times N^2 + \left(\frac{25}{6} + NB\right) \times N \text{ additions}$$

$$\frac{N^3}{6} + \left(\frac{13}{4} + NB\right) \times N^2 + \left(\frac{5}{6} + NB\right) \times N \text{ multiplications}$$

$$\frac{N^2}{2} + \left(\frac{5}{2} + NB\right) \times N \text{ divisions}$$

at most $N^2 + N$ comparisons

SYSS — symmetric linear system solution with condition estimation

Purpose: SYSS (SYmmetric System Solution) solves the system AX = B where A is a symmetric matrix. It also provides an estimate of the condition number of A. A does not have to be positive definite.

Usage: CALL SYSS (N, C, B, IB, NB, COND)

N → the number of equations

C → a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

C is overwritten during the solution.

B → the matrix of right-hand sides, dimensioned (IB, KB) in the calling program, where IB≥N and KB≥NB

← the solution X

IB → the row (leading) dimension of B, as dimensioned in the calling program

NB → the number of right-hand sides

COND ← an estimate of the condition number of A (see **Note 1**)

Note 1: The condition number measures the sensitivity of the solution of a linear system to errors in the matrix and in the right-hand side. If the elements of the matrix and the right-hand side(s) of your linear system have **d** decimal digits of precision, the solution might have as few as **d** $-\log_{10}$(COND) correct decimal digits. Thus if COND is greater than $10^{\mathrm{Bd}P}$, there may be no correct digits.

If the given matrix, A, is known in advance to be well-conditioned, then the user may wish to use the routine SYLE, which is a little faster than SYSS. Ordinarily, however, the user is strongly urged to choose SYSS, and to follow it by a test of the condition estimate.

```
C
       CALL MOVEFR(N,X,B)
C
C SOLVE THE SYSTEM
C
       CALL SYLE(N,C,B,N,1)
C
C COMPUTE THE RELATIVE ERROR AND THE RELATIVE RESIDUAL
C
       CALL SYML(N,CC,B,R)
       ERR=0.0
       DO 30 I=1,N
          ERR=AMAX1(ERR,ABS(B(I)-FLOAT(I)))
          R(I)=R(I)-X(I)
 30    CONTINUE
       XNORM=SAMAX(N,X,1)
       RNORM=SAMAX(N,R,1)
       RELERR=ERR/XNORM
       RELRES=RNORM/(XNORM*SYNM(N,CC))
       IWRITE=I1MACH(2)
       WRITE(IWRITE,31)RELERR,RELRES
 31    FORMAT(16H RELATIVE ERROR=,E15.5,19H RELATIVE RESIDUAL=,
     1  E15.5)
       STOP
       END
```

When the above program was executed on the Honeywell 6000 machine at Bell Laboratories, the following was printed:

```
RELATIVE ERROR=  0.10544E-07 RELATIVE RESIDUAL=  0.95702E-11
```

The condition number of the matrix (see the example in SYSS) is about 1300, and the machine precision on the Honeywell computer is about 10^{-8}. Thus even in the absence of roundoff error in SYML, a relative error of 1.3×10^{-5} would not be surprising. The relative error given above is quite within reason. The relative residual, as promised, satisfies (1.1) even though the problem is slightly ill-conditioned.

Time: N^2 additions
N comparisons

See also: SYDC, SYMD, SYLE, SYSS, SYCE

Author: Linda Kaufman

Example: The subroutines in the library for solving $Ax = b$ are designed to return computed solutions x such that the residual $r = Ax - b$ satifies

$$\frac{||r||}{||A|| \; ||x||} \le \varepsilon \qquad (1.1)$$

where ε is the machine precision. In this example we show that if A is ill-condtioned, then the computed solution need not be calose to the true solution even though equation (1.1) is satisfied. The subroutine SYNM is used to compute the left-hand side of (1.1). The matrix in this example is given by

$$a_{ij} = | i - j |$$

and the true solution is $x_i = i$. The right hand side is generated using SYML and the computed solution is obtained using SYLE. The subroutine SAMAX is used to compute the 1-norm of a vector.i.e. $\max_{1 \le i \le n} |x_i|$

```
      INTEGER I, J, L, N, I1MACH, IWRITE
      REAL C(1300), CC(1300), B(50), X(50)
      REAL RELERR, RELRES, XNORM, RNORM, ERR, R(50)
      REAL SYNM, SAMAX
      L=0
C
C GENERATE MATRIX
C
      N=50
      DO 20 I=1,N
         DO 10 J=I,N
            L=L+1
            C(L)=J-I
            CC(L)=C(L)
 10      CONTINUE
         B(I)=I
 20   CONTINUE
C
C GENERATE RIGHT HAND SIDE
C
      CALL SYML(N,C,B,X)
C
C MAKE COPY OF RIGHT HAND SIDE
```

SYNM — norm of a symmetric matrix

Purpose: SYNM (SYmmetric matrix NorM) computes the norm of a symmetric matrix A stored in packed form. The infinity norm is defined as $\max_{1 \le i \le n} \sum_{j=1}^{n} | a_{ij} |$

Type: Real function

Usage: <answer> = SYNM (N, C)

N $\longrightarrow$ the number of rows in A

C $\longrightarrow$ a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

<answer> $\longleftarrow$ $\max_{1 \le i \le n} \sum_{j=1}^{n} | a_{ij} |$

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1

Double precision version: DSYNM with C and DSYNM declared double precision

Complex version: CSYNM with C declared complex

Storage: None

```
      CALL SYLE(N,C,B,N,1)
C
C PRINT THE COMPUTED AND TRUE SOLUTION
C
      IWRITE=I1MACH(2)
      WRITE(IWRITE,31)
 31   FORMAT(34H TRUE SOLUTION   COMPUTED SOLUTION)
      WRITE(IWRITE,32)(X(I),B(I),I=1,N)
 32   FORMAT(1H ,2E17.8)
C
C COMPUTE THE RELATIVE ERROR
C
      ERR=0.0
      DO 40 I=1,N
         ERR=ERR+ABS(B(I)-X(I))
 40   CONTINUE
      ERR=ERR/SASUM(N,X,1)
      WRITE(IWRITE,41)ERR
 41   FORMAT(19H RELATIVE ERROR IS ,1PE15.7)
      STOP
      END
```

When the above program was executed on the Honeywell 6000 machine at Bell Laboratories, the following was printed:

```
   TRUE SOLUTION         COMPUTED SOLUTION
   0.22925607E 00          0.22925607E 00
   0.76687502E 00          0.76687498E 00
   0.68317685E 00          0.68317696E 00
   0.50919111E 00          0.50919100E 00
   0.87455959E 00          0.87455969E 00
   0.64464101E 00          0.64464090E 00
   0.84746840E 00          0.84746833E 00
   0.35396343E 00          0.35396342E 00
   0.39889160E 00          0.39889174E 00
   0.45709422E 00          0.45709418E 00
RELATIVE ERROR IS   1.2794319E-07
```

The condition number of the matrix (see the example in SYSS) is about 54. and the machine precision, on the Honeywell computer is about 10^{-8}. Thus even in the absence of roundoff error in SYML, a relative error of 5×10^{-7} would not be surprising. The value computed above is quite reasonable.

See also: SYFBS, SYCE, SYDC, SYMD, SYLE, SYSS

Author: Linda Kaufman

Example: This example checks the consistency of SYML and SYLE, the symmetric linear equation solver.

First the example uses SYML to compute for a given vector x and matrix A, the vector

$$b = Ax.$$

Then the problem is inverted, i.e., SYLE is used to find the vector x which satisfies

$$Ax = b$$

This x is then compared with the original vector. The 10×10 symmetric matrix A is chosen so that

$$a_{ij} = |i-j|.$$

The vector x is chosen randomly.

```
        INTEGER N, L, I, J, IWRITE, I1MACH
        REAL C(55), X(10), B(10)
        REAL UNI, ERR, SASUM, ABS
        N=10
C
C CONSTRUCT THE MATRIX A(I,J)=ABS(J-I) AND PACK INTO C
C
        L=0
        DO 20 I=1,N
           DO 10 J=I,N
              L=L+1
              C(L)=J-I
  10       CONTINUE
  20    CONTINUE
C
C CONSTRUCT A RANDOM VECTOR X
C
        DO 30 I=1,N
           X(I)=UNI(0)
  30    CONTINUE
C
C FIND THE VECTOR B=AX
C
       CALL SYML(N,C,X,B)
C
C SOLVE THE SYSTEM AX=B
C
```

SYML — symmetric matrix - vector multiplication

Purpose: SYML (SYmmetric matrix MuLtiplication) forms the product *Ax* where *A* is a general symmetric matrix stored in packed form.

Usage: CALL SYML(N, C, X, B)

N $\longrightarrow$ the length of *x*

C $\longrightarrow$ a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

X $\longrightarrow$ the vector *x* to be multiplied

B $\longleftarrow$ the vector A*x*

Error situations: (All errors in this subprogram are fatal — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1

Double-precision version: DSYML with C, X, and B declared double precision.

Complex version: CSYML with C, X, and B declared complex

Complex Hermitian version: CHEML with C, X, and B declared complex

Time: N^2 additions
N^2 multiplications

Reference: Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

Example: The following program fragment determines whether a matrix is positive definite. According to the theory given in the reference above, a symmetric matrix is positive definite only if D in the decomposition computed by SYMD is diagonal with positive diagonal elements. If D is diagonal, all the elements of INTER are positive and the elements of D are packed into C in the same positions that the diagonal of A had originally occupied.

```
       CALL SYMD(N,C,INTER,0.0)
       IWRITE=I1MACH(2)
C
C DETERMINE IF THE MATRIX PACKED INTO C IS POSITIVE DEFINITE.
C THE INDEX K PICKS OUT THE DIAGONAL OF THE MATRIX D
C OF THE DECOMPOSITION
        K=1
        DO 10 I=1,N
           IF(INTER(I).LT.0.OR.C(K).LE.0.0) GO TO 20
C FIND NEXT DIAGONAL ELEMENT
           K = K + N - I
10      CONTINUE
        WRITE(IWRITE,11)
11      FORMAT(32H THE MATRIX IS POSITIVE DEFINITE)
        GO TO 30
20      WRITE(IWRITE,21)
21      FORMAT(36H THE MATRIX IS NOT POSITIVE DEFINITE)
30      CONTINUE
```

Error situations: *(The user can elect to 'recover' from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYMD with C and EPS declared double precision.

Complex symmetric version: CSYMD with C declared complex

Complex Hermitian version: CHEMD with C declared complex (see **Note 2**).

Storage: None

Time:
$\frac{N^3}{6} + \frac{N^2}{4} + \frac{7}{6}N$ additions
$\frac{N^3}{6} + \frac{N^2}{4} + \frac{11}{6}N$ additions
$\frac{N^2}{2} + \frac{N}{2}$ divisions
at most $N^2 - 1$ comparisons

Method: The Bunch - Kaufman algorithm, described in the reference below, is used.

See also: SYLE, SYDC, SYFBS, SYSS, SYCE

Author: Linda Kaufman

SYMD — MDM^T decomposition of a symmetric matrix

Purpose: SYMD (SYmmetric MDM^T decomposition) forms the decomposition $PMDM^TP^T$ of a symmetric matrix A, where P is a permutation matrix, M is unit lower triangular matrix, and D is block diagonal. The matrix A need not be positive definite. This subroutine allows the user to specify a threshold for considering the matrix singular. It is called by the decomposition routines SYDC and SYCE.

Usage: CALL SYMD (N, C, INTER, EPS)

N $\longrightarrow$ the order of the matrix A

C $\longrightarrow$ a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

$\longleftarrow$ the D and M matrices of the decomposition for SYFBS (see **Note 1**)

INTER $\longleftarrow$ a vector of length N containing a record of the interchanges, i.e. the matrix P, described in **Note 1** below.

EPS $\longrightarrow$ if $|\, d_{kk} \,| \le$ EPS and d_{kk} corresponds to a 1×1 block of D, then C is considered a singular matrix whose rank is at least k–1

Note 1: The MDM^T decomposition of a symmetric matrix A satisfies $P^TAP = MDM^T$ where P is a permutation matrix, M is a unit lower triangular matrix, and D is a block diagonal matrix with blocks of order 1×1 and blocks of order 2×2. Whenever $d_{i+1,i}$ is nonzero (in a 2×2 block of D), $m_{i+1,i}$ is zero. On return from SYMD, d_{ii}, the diagonal of D, occupies the position of C which contained a_{ii} on entry, and the elements of the strictly lower portions of M and D appear permuted in the remaining positions of C. Since the diagonal elements of M are all 1, they are not stored. The positive elements of INTER contain information for constructing P (see the introduction to this chapter). The negative elements of INTER, if any, indicate the presence of 2×2 blocks in D. If INTER(I) is negative, D contains a 2×2 block beginning at row I–1. In this case, $d_{i,i-1}$ directly follows $d_{i-1,i-1}$ in C.

Note 2: For complex Hermitian matrices(A = A*, where A* is the conjugate transpose of A), the complex Hermitian version of this subroutine computes the MDM* decomposition and returns the conjugate of M rather than M in C.

Examples:

1. The following call to SYLE replaces the $N \times K$ matrix B with A^{-1} B where A is a symmetric matrix packed into a vector C. Note that A^{-1} is not computed

```
CALL SYLE (N, C, B, N, K)
```

2. On the other hand, to compute the inverse A^{-1}, one may set B to the identity matrix and call SYLE. The following program fragment leaves A^{-1} in the matrix B. It does not use the fact that A^{-1} is symmetric.

```
      DO 20 I=1,N
         DO 10 J=1,N
            B(I,J)=0.0
10       CONTINUE
         B(I,I)=1.0
20    CONTINUE
      CALL SYLE(N, C, B, N, N)
```

Error situations: *(The user can elect to ‘recover’ from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1
2	IB < N
3	NB < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYLE with C and B declared double precision.

Complex symmetric version: CSYLE with C and B declared complex

Complex Hermitian version: CHELE with C and B declared complex

Storage: N integer locations of scratch storage in the dynamic storage stack

Time:

$$\frac{N^3}{6} + (\frac{3}{4} + NB) \times N^2 + (\frac{7}{6} + NB)\, N \text{ additions}$$

$$\frac{N^3}{6} + (\frac{1}{4} + NB) \times N^2 + (\frac{11}{6} + NB) \times N \text{ multiplications}$$

$$\frac{N^2}{2} + (\frac{1}{2} + NB) \times N \text{ divisions}$$

at most $N^2 - 1$ comparisons

Method: The Bunch - Kaufman algorithm, described in the reference below, is used.

See also: SYDC, SYFBS, SYMD, SYSS, SYCE

Author: Linda Kaufman

Reference:
Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

SYLE — symmetric linear system solution

Purpose: SYLE (SYmmetric Linear Equation solution) solves AX = B where A is a symmetric matrix. A does not have to be positive definite.

Usage: CALL SYLE (N, C, B, IB, NB)

N $\longrightarrow$ the number of equations

C $\longrightarrow$ a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

C is overwritten during the solution.

B $\longrightarrow$ the matrix of right-hand sides, dimensioned (IB,KB) in the calling program, where IB≥N and KB≥NB

$\longleftarrow$ the solution X

IB $\longrightarrow$ the row (leading) dimension of B, as dimensioned in the calling program

NB $\longrightarrow$ the number of right-hand sides

Note 1: Unless the given matrix A, is known in advance to be well-conditioned, the user should use the routine SYSS instead of SYLE.

Note 2: Users who wish to solve a sequence of problems with the same coefficient matrix, but different right-hand sides *not all known in advance*, should not use SYLE, but should call subprograms SYDC and SYFBS. (See the example in SYCE.) SYDC is called once to get the MDM^T decomposition (see the introduction to this chapter) and then SYFBS is called for each new right-hand side.

Example: The program fragment below replaces the K×N matrix B with BA^{-1} where A is a symmetric matrix packed into C according to the scheme given in the parameter list of SYDC. Note that A^{-1} is not formed explicitly since forming BA^{-1} is equivalent to solving XA = B for the matrix X. Because A is symmetric, solving XA = B is, in turn, equivalent to solving $AX^T = B^T$. Thus the problem reduces to solving a linear system with ‘K’ right-hand sides, each of which unfortunately resides in a ‘row’ of the array B, rather than a column of an array. In the program fragment we chose not to transpose the matrix B but to invoke SYFBS K times and store each row of B temporarily in the one-dimensional array Y.

```
          CALL SYDC(N,C,INTER)
          DO 30 I=1,K
             DO 10 J=1,N
                Y(J)=B(I,J)
10           CONTINUE
             CALL SYFBS(N,C,Y,N,1,INTER)
             DO 20 J=1,N
                B(I,J)=Y(J)
20           CONTINUE
30        CONTINUE
```

Complex version: CSYFBS with C and B declared complex

Complex Hermitian version: CHEFBS with C and B declared complex

Storage: None

Time: $NB \times (N^2 - N)$ additions
$NB \times (N^2 + N)$ multiplications
$NB \times N$ divisions

Method: The Bunch - Kaufman algorithm, described in the reference below, is used.

See also: SYLE, SYDC, SYMD, SYSS, SYCE

Author: Linda Kaufman

Reference: Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

SYFBS — forward and back solve for symmetric matrices

Purpose: SYFBS (SYmmetric matrix Forward and Back Solution) solves AX = B where A is a symmetric matrix using the decomposition of A computed by SYCE, SYDC, or SYMD. It is called by both SYSS and SYLE to solve symmetric linear systems.

Usage: CALL SYFBS (N, C, B, IB, NB, INTER)

N	→	the number of equations
C	→	a one-dimensional array of length N(N+1)/2 containing the matrices M and D computed by SYDC, SYCE, or SYMD
B	→	the matrix of right-hand sides, dimensioned (IB,KB) in the calling program, where IB≥N and KB≥NB
	←	the solution X
IB	→	the row (leading) dimension of B, as dimensioned in the calling program
NB	→	the number of right-hand sides
INTER	→	an integer vector of length N containing a record of the interchanges performed by SYDC, SYCE, or SYMD

Error situations: *(The user can elect to ‘recover’ from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1
2	IB < N
3	NB < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYFBS with C and B declared double precision.

Example: The program fragment below determines how many positive eigenvalues a symmetric matrix, A, has.

According to the reference above the signs of the eigenvalues of A correspond to the signs of the eigenvalues of D in the MDM^T decomposition of A. Moreover, each 2×2 block in D corresponds to a positive-negative eigenvalue pair. Thus the number of positive eigenvalues of D is equal to the number of 2×2 blocks of D plus the number of positive 1×1 blocks.

The program first counts the number of 2×2 blocks of D by counting the number of negative elements in the array INTER, since each negative element (see **Note 1** above) signals the existence of a 2×2 block. It adds to this count the number of positive 1×1 blocks of D in order to find the total number of positive eigenvalues of A.

```
        CALL SYDC(N,C,INTER)
        NPOS=0
C
C COUNT THE NUMBER OF POSITIVE EIGENVALUES OF D AND HENCE
C OF THE MATRIX WHICH HAD ORIGINALLY BEEN PACKED INTO C
C
C THE INDEX K PICKS OUT THE DIAGONAL OF THE MATRIX D OF
C THE DECOMPOSITION
C
        K=1
        I=1
  10    IF (I-N) 20,30,40
  20    (INTER(I+1) .GT. 0) GO TO 30
C
C OTHERWISE WE HAVE A 2×2 BLOCK
C
          NPOS=NPOS+1
          K=K+2*(N-I)+1
          I=I+2
          GO TO 10
C
C WE HAVE A 1×1 BLOCK
C
  30      IF(C(K) .GT. 0.0) NPOS=NPOS+1
          K = K + N - I
          I=I+1
          GO TO 10
  40    IWRITE=I1MACH(2)
        WRITE(IWRITE,41)NPOS
  41    FORMAT(38H THE NUMBER OF POSITIVE EIGENVALUES IS,I5)
```

Error situations: *(The user can elect to ‘recover’ from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Number	Error
1	N < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYDC with C declared double precision.

Complex symmetric version: CSYDC with C declared complex

Complex Hermitian version: CHEDC with C declared complex (see **Note 2** above).

Storage: None

Time: $\frac{N^3}{6} + \frac{3}{4}N^2 + \frac{7}{6}N$ additions

$\frac{N^3}{6} + \frac{N^2}{4} + \frac{11}{6}N$ multiplications

$\frac{N^2}{2} + \frac{N}{2}$ divisions

at most $N^2 - 1$ comparisons

Method: The Bunch - Kaufman algorithm, described in reference [1] below, is used.

SYDC calls SYMD after setting EPS = $||A||\ \varepsilon$, where ε is the machine precision, i.e. the value returned by R1MACH(4) (or, for double precision, by D1MACH(4)).

See also: SYLE, SYFBS, SYMD, SYCE, SYSS

Author: Linda Kaufman

Reference: Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

SYDC — decomposition of a symmetric matrix

Purpose: SYDC (SYmmetric matrix DeComposition) forms the MDM^T decomposition of a symmetric matrix A, which need not be positive definite. It is called by SYLE as the first step of the solution of a symmetric linear system.

Usage: CALL SYDC (N, C, INTER)

N $\longrightarrow$ the order of the matrix A

C $\longrightarrow$ a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

$\longleftarrow$ the D and M matrices of the decomposition for SYFBS (see **Note 1**)

INTER $\longleftarrow$ an integer vector of length N containing a record of the interchanges, i.e. the matrix P, described in **Note 1** below.

Note 1: The MDM^T decomposition of a symmetric matrix A satisfies $P^TAP = MDM^T$ where P is a permutation matrix, M is a unit lower triangular matrix, and D is a block diagonal matrix with blocks of order 1×1 and blocks of order 2×2. Whenever $d_{i+1,i}$ is nonzero (in a 2×2 block of D), $m_{i+1,i}$ is zero. On return from SYDC, d_{ii}, the diagonal of D, occupies the position of C which contained a_{ii} on entry, and the elements of the strictly lower portions of M and D appear permuted in the remaining positions of C. Since the diagonal elements of M are all 1, they are not stored. The positive elements of INTER contain information for constructing P (see the introduction to this chapter). The negative elements of INTER, if any, indicate the presence of 2×2 blocks in D. If INTER(I) is negative, D contains a 2×2 block beginning at row I–1. In this case, $d_{i,i-1}$ directly follows $d_{i-1,i-1}$ in C.

Note 2: For complex Hermitian matrices (A = A*, where A* is the conjugate transpose of A), the complex Hermitian version of this subroutine computes the MDM* decomposition and returns the conjugate of M rather than M in C.

the following results were obtained on the Honeywell 6000 computer at Bell Labs:

```
CONDITION NUMBER IS   6.71523875E 05
THE FIRST SOLUTION X, FROM SYCE AND SYFBS=
         -8.0002890
         -3.0003689
         -1.9998967
         -5.0000131
          8.0000029
THE SOLUTION AFTER ITERATION     1
         -8.0000000
         -3.0000000
         -2.0000000
         -5.0000000
          8.0000000
THE SOLUTION AFTER ITERATION     2
         -8.0000000
         -3.0000000
         -2.0000000
         -5.0000000
          8.0000000
```

As in most iterative algorithms, the algorithm implemented above stops when the change in the solution is sufficiently small. Although the solution at the end of iteration 1 is correct, the change from the original solution was large and hence the program decided to take one more step.

The first solution above is inaccurate, as would have been expected from the estimate of the condition number for the matrix. The iterative refinement algorithm successfully improved the solution to this problem because the matrix and the right-hand side could be exactly represented in the machine. (Also the condition number was not high.) Often the input matrix cannot be represented exactly and the iterative refinement algorithm produces a very accurate, but worthless, solution to a slightly incorrect problem.

```
  50         D(I)=DBLE(SAVEB(I))
          L=1
          DO 70 I=1,N
             DO 60 J=I,N
                IF (I.NE.J) D(J)=D(J) - DBLE(SAVEC(L))*B(I)
                D(I) = D(I) - DBLE(SAVEC(L))*B(J)
  60         L=L+1
             R(I) = D(I)
  70      CONTINUE
C
C SOLVE A(DELTAX) =R
C
          CALL SYFBS(N,C,R,8,1,INTER)
C
C DETERMINE NORM OF CORRECTION AND ADD IN CORRECTION
C
          WRITE(IWRITE,71)ITER
  71      FORMAT(30H THE SOLUTION AFTER ITERATION ,I5)
          RNORM=0.0
          DO 80 I=1,N
                B(I) = B(I) + R(I)
                RNORM=RNORM+ABS(R(I))
                WRITE(IWRITE,41)B(I)
  80      CONTINUE
       IF(RNORM.LT.R1MACH(4)*BNORM) STOP
  90   CONTINUE
       WRITE(IWRITE,91)
  91   FORMAT(29H ITERATIVE IMPROVEMENT FAILED)
       STOP
       END
```

The above program was applied to a problem in which the upper triangular portion of the symmetric matrix A was given by

–4.0	0.0	–16.	–32.	28.0
	1.0	5.0	10.0	–6.0
		–37.0	–66.0	64.0
			–85.0	53.0
				–15.0

Of course when the matrix was read in to the C array, to conform to FORTRAN conventions each of the above lines had to be left justified.

When the following right hand side was read in

448.
–111.
1029.
1207.
–719.

```
      INTEGER N, JEND, IREAD, I1MACH, I, JBEGIN, J, IWRITE
      INTEGER INTER(6), IEND, ITER, L, IFIX
      REAL C(20), SAVEC(36), B(6), SAVEB(6), R(6)
      REAL COND, R1MACH, BNORM, RNORM, ABS, ALOG10
      DOUBLE PRECISION D(6)
      N=5
C
C READ IN A SYMMETRIC MATRIX WHOSE UPPER TRIANGULAR
C PORTION IS STORED ONE ROW PER CARD. MAKE A
C COPY OF THE MATRIX SO THAT IT CAN BE USED LATER
C
       JEND=0
       IREAD=I1MACH(1)
       DO 20 I=1,N
          JBEGIN=JEND+1
          JEND=JBEGIN+N - I
          READ(IREAD,1)(C(J),J=JBEGIN,JEND)
  1       FORMAT(5F8.0)
          DO 10 J=JBEGIN,JEND
               SAVEC(J)=C(J)
  10      CONTINUE
  20   CONTINUE
C READ IN RIGHT HAND SIDE AND SAVE IT
       DO 30 I=1,N
          READ(IREAD,1)B(I)
          SAVEB(I)=B(I)
  30   CONTINUE
C
C  SOLVE AX = B USING SEPARATE CALLS TO SYCE AND SYFBS
C
      CALL SYCE(N,C,INTER,COND)
      CALL SYFBS(N,C,B,6,1,INTER)
      IWRITE=I1MACH(2)
      IF(COND.GE.1.0/R1MACH(4))WRITE(IWRITE,31)
  31  FORMAT(49H CONDITION NUMBER HIGH,ACCURATE SOLUTION UNLIKELY)
      WRITE(IWRITE,32) COND
  32  FORMAT(21H CONDITION NUMBER IS ,1PE16.8)
C     COMPUTE NORM OF SOLUTION
      BNORM=0.0
      WRITE(IWRITE,33)
  33  FORMAT(43H THE FIRST SOLUTION X, FROM SYCE AND SYFBS=)
      DO 40 I=1,N
         BNORM=BNORM+ABS(B(I))
  40     WRITE(IWRITE,41)B(I)
  41  FORMAT(1H ,F20.7)
C
C IEND IS THE UPPER BOUND ON THE NUMBER OF BITS PER WORD
C
      IEND=I1MACH(11)*IFIX(R1MACH(5)/ALOG10(2.0)+1.0)
C
C REFINE SOLUTION
C
      DO 90 ITER=1,IEND
C
C COMPUTE RESIDUAL R = B - AX, IN DOUBLE PRECISION
C
         DO 50 I=1,N
```

References: [1] Bunch, J. R., Kaufman, L., and Parlett, B., Decomposition of a symmetric matrix, *Numer. Math 27* (1976), 95-109.

[2] Cline, A. K., Moler, C. B., Stewart, G. W., and Wilkinson, J. H., An estimate for the condition number, *SIAM J. Numer. Anal. 16* (1979), 368-375.

Example: This example is an encoding of the iterative refinement algorithm which may be used to obtain a highly accurate solution to a system of linear equations with an ill-conditioned coefficient matrix. If the condition number is not excessively high, the program usually returns a solution that is accurate to the working precision of the machine.

The iterative refinement algorithm is essentially:

(1) Solve $Ax = b$
(2) Set $tol = \varepsilon \sum |x_i|$
where ε is the precision of the machine
(3) Compute in double precision the residual
$r = Ax - b$
(4) Solve $A\,\delta x = r$
(5) Compute $norm = \sum |\delta x_i|$
(6) Set x to $x + \delta x$
(7) If $norm \le tol$ stop, else return to step 3

In the program below, step (1) is accomplished using the two lower-level subroutines SYCE and SYFBS. SYCE decomposes A into several factors and SYFBS solves the system using these factors. Since A is destroyed by SYCE and needed in step (3) of the algorithm, a copy of the A matrix is saved. In step (4) the decomposition created earlier in SYCE is reused and only the forward and back solver SYFBS is required. Since it is possible that the matrix is so ill-conditioned that the iterative refinement algorithm will diverge, steps (3) through (7) in the program are performed only a finite number of times. This number is chosen to be an upper bound on the number of bits in the mantissa of the floating-point number supported by the machine.

This algorithm is not included in PORT because for double-precision matrices part of the computation would have to be done in extended precision.

Number	Error
1	N < 1
10 + k*	singular matrix whose rank is at least k

Double-precision version: DSYCE with C declared double precision.

Complex symmetric version: CSYCE with C declared complex

Complex Hermitian version: CHECE with C declared complex (see **Note 3**).

Storage: N real (double precision for DSYCE, complex for CSYCE) locations of scratch storage in the dynamic storage stack

Time: $\frac{N^3}{6} + \frac{19}{4}N^2 + \frac{25}{6}N$ additions

$\frac{N^3}{6} + \frac{13}{4}N^2 + \frac{5}{6}N$ multiplications

$\frac{N^2}{2} + \frac{5}{2}N$ divisions

at most $N^2 + N$ comparisons

Method: The Bunch - Kaufman algorithm, described in reference [1] below, is used to determine the decomposition. The algorithm in [2] is used to get the condition estimate.

SYCE calls SYMD after setting EPS to 0.0

See also: SYLE, SYFBS, SYMD, SYDC, SYSS

Author: Linda Kaufman

Purpose: SYCE (SYmmetric matrix Condition Estimation) gives a lower bound for the condition number of a symmetric matrix A, which need not be positive definite. It also supplies the MDM^T decomposition of A and may be used in a linear equation package.

Usage: CALL SYCE (N, C, INTER, COND)

N → the number of rows in A

C → a one-dimensional array of length N(N+1)/2 into which the lower triangular part of the matrix A is packed by columns as illustrated in the following 4×4 example:

$$\begin{matrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix} \quad \longrightarrow \quad \begin{matrix} c_1 & & & \\ c_2 & c_5 & & \\ c_3 & c_6 & c_8 & \\ c_4 & c_7 & c_9 & c_{10} \end{matrix}$$

← the D and M matrices of the decomposition for SYFBS (see **Note 2**)

INTER ← an integer vector of length N containing a record of the interchanges, i.e. the matrix P, described in **Note 2** below.

COND ← an estimate of the condition number of A (see **Note 1**)

Note 1: The condition number measures the sensitivity of the solution of a linear system to errors in the matrix and in the right-hand side. If the elements of the matrix and the right-hand side(s) of your linear system have **d** decimal digits of precision, the solution might have as few as **d** $-\log_{10}$(COND) correct decimal digits. Thus if COND is greater than $10^{\mathrm{Bd}P}$, there may be no correct digits.

Note 2: The MDM^T decomposition of a symmetric matrix A satisfies $P^T AP = \mathrm{MDM}^T$, where P is a permutation matrix, M is a unit lower triangular matrix, and D is a block diagonal matrix with blocks of order 1×1 and blocks of order 2×2. Whenever $d_{i+1,i}$ is nonzero (in a 2×2 block of D), $m_{i+1,i}$ is zero. On return from SYCE, d_{ii}, the diagonal of D, occupies the position of C which contained a_{ii} on entry, and the elements of the strictly lower portions of M and D appear permuted in the remaining positions of C. Since the diagonal elements of M are all 1, they are not stored. The positive elements of INTER contain information for constructing P (see the introduction to this chapter). The negative elements of INTER, if any, indicate the presence of 2 × 2 blocks in D. If INTER(I) is negative, then D contains a 2×2 block beginning at row I–1. In this case, $d_{i,i-1}$ directly follows $d_{i-1,i-1}$ in C.

Note 3: For complex Hermitian matrices (A = A*, where A* is the conjugate transpose of A), the complex Hermitian version of this subroutine computes the MDM* decomposition and returns the conjugate of M rather than M in C.

Error situations: *(The user can elect to ‘recover’ from those errors marked with an asterisk — see *Error Handling*, Framework Chapter)

Appendix 2

SYMMETRIC MATRICES

SYCE - Condition Estimation
SYDC - DeComposition
SYFBS - Forward and Back Solve
SYLE - Linear Equation solution
SYMD - MDM^T decomposition
SYML - MuLtiplication
SYNM - NorM
SYSS - System Solution