INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-289

SSLE - A FORTRAN PACKAGE FOR SOLVING SPARSE LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

April 1982

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

No. SOC-289

SSLE - A FORTRAN PACKAGE FOR SOLVING SPARSE LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

April 1982

SSLE - A FORTRAN PACKAGE FOR SOLVING SPARSE LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

Abstract

SSLE is a package of subroutines for solving systems of sparse linear equations. The bi-factorization method is used for systems having a symmetrical structure of the coefficient matrix. The method is based on Zollenkopf's algorithm. In the case of a nonsymmetrical structure, the necessary zero elements are added. There are four versions of the principal subroutine appropriate for a symmetrical or nonsymmetrical as well as for a real or complex coefficient matrix.

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant G0647.

J.A. Starzyk and J.W. Bandler are with the Group on Simulation, Optimization and Control and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

J.A. Starzyk is on leave from the Institute of Electronics Fundamentals, Technical University of Warsaw, Poland.

I. INTRODUCTION

SSLE is a package of subroutines for solving systems of sparse linear equations. Its main features include a 1971 version of the Zollenkopf bi-factorization algorithm [1].

The bi-factorization method should be used for sparse coefficient matrices that have non-zero diagonal terms and are either strictly symmetric or asymmetric in element value but with a symmetric sparsity structure. Furthermore, it is assumed for reasons of round-off error that the matrix is either symmetric and positive definite or is diagonally dominant (a matrix is diagonally dominant by rows if each diagonal element is not less than the sum of the moduli of the other elements in its row; a similar definition holds for diagonal dominance by columns).

The package is written in Fortran IV, documented and tested on the CDC 170/730 (System B) at McMaster University. It is available as a permanent group file in the form of a library of binary relocatable subroutines. The name of the library is SSLE.

II. SUBROUTINE AND VARIABLES

This section describes all the subroutines and variables that could be of interest to the user.

There are four principal subroutines called by the user's main program:

(1) SSLERS - for a symmetrical, real coefficient matrix

(2) SSLERN - for a nonsymmetrical, real coefficient matrix

(3) SSLECS - for a symmetrical, complex coefficient matrix

(4) SSLECN - for a nonsymmetrical, complex coefficient matrix

These subroutines have a similar structure as shown in Fig. 1. Names of blocks from Fig. 1 are presented in Table 1.

Table 1 Subroutines in the library SSLE

	1	2	3	4
MAIN	SSLERS	SSLERN	SSLECS	SSLECN
A	DATARS	DATARN	DATACS	DATACN
B	DATBRS	DATBRN	DATBCS	DATBCN
C	SORDRS	SORDRN	SORDCS	SORDCN
D	REDURS	REDURN	REDUCS	REDUCN
E	SOLVRS	SOLVRN	SOLVCS	SOLVCN
F	SYMCOLR		SYMCOLC	

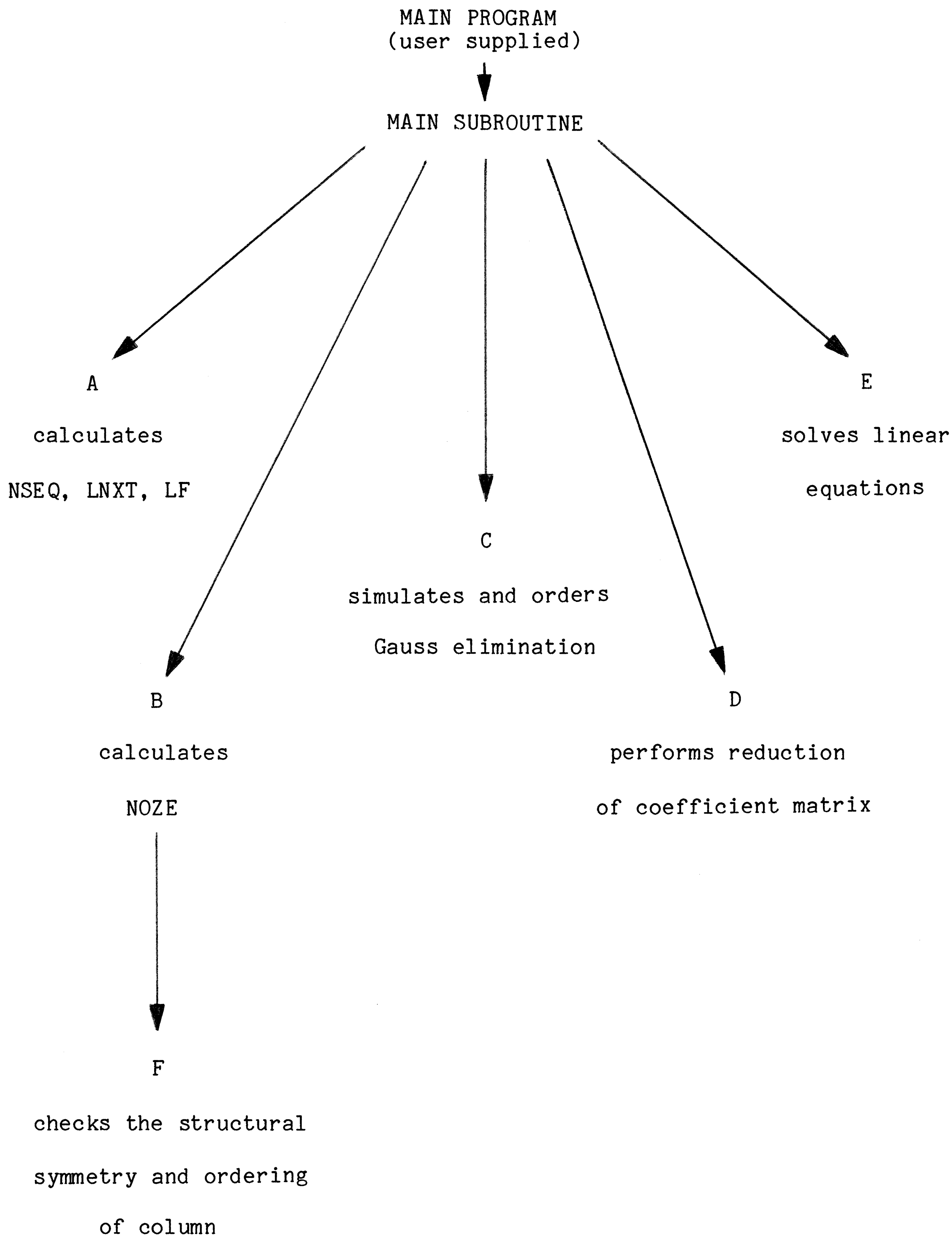

Fig. 1. Structure of subroutines solving linear equations.

Integer Variables

N — Number of unknowns, order of the coefficient matrix.

LCOL — An array of dimension N+1 pointing to the starting position of the columns. (N+1)-st element of LCOL indicates the first vacant position.

NOZE — An array of dimension N storing the number of nonzero terms in each column of the CE matrix. This array is calculated by the library subroutines.

NSEQ — An array of dimension N identifying the sequence of pivotal indices. This array is calculated by the library subroutines.

ITAG — An array of dimension LES storing row indices of elements stored in the matrix CE.

LNXT — An array of dimension LES storing the location of the next term in each column. This array is calculated by the library subroutines.

LF — Indicator for the first vacant position.

LES — Dimension of matrices ITAG, LNXT, CE and RE.

IAR — Indicator for checking the structural symmetry and column ordering of the coefficient matrix as well as the flag for insufficient area.

Complex Variables

CE — An array of dimension LES having columnwise stored terms of the coefficient matrix.

RE — An array of dimension LES having rowwise stored off-diagonal terms of the coefficient matrix. This array is calculated by the library subroutines.

DE An array of dimension N storing diagonal terms of the coefficient matrix.

V An array of dimension N storing the right hand side vector. At the output V contains the solution vector.

The length of the arrays LCOL, NOZE, NSEQ, DE and V are well-defined by the number of equations. The dimension of the arrays ITAG, LNXT, RE and CE, however, cannot be determined in advance. Thus, it is recommended to reserve sufficient memory space (2 to 5 times more than the number of nonzero elements).

The vacant positions of array CE and the last position of table LNXT must be set to zero. The other vacant positions of LNXT must be numbered consecutively.

III. HOW TO USE THE PROGRAM

In order to solve a sparse system of linear equations, the user has to (i) prepare the main program, (ii) prepare file or cards which contain all the data.

The main program should read all the necessary data and execute the call to subroutines SSLERS, SSLERN, SSLECS or SSLECN depending on the type of equations to be solved. The main program should also provide the printing output for the solution stored in vector V. All the arrays should be suitably dimensioned in the main program.

The subroutines are called in the following form:

```
SSLERS(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,V,LF,LES,IAR)
SSLERN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR)
SSLECS(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,V,LF,LES,IAR)
SSLECN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR)
```

Because the form of data is different in the case of symmetrical and nonsymmetrical coefficient matrix we will discuss them separately.

Symmetrical Systems

The user should enter the following variables.

N number of equations.

CE contains columnwise stored terms of the coefficient matrix (together with diagonal elements).

LCOL starting positions of columns in matrix CE; LCOL(N+1) indicates the first vacant position and is equal to the number of nonzero elements +1.

ITAG,LES and V.

Nonsymmetrical Systems

The user should enter the following variables.

N number of equations.

CE contains columnwise stored off-diagonal terms of the coefficient matrix.

LCOL starting positions of columns in matrix CE; LCOL(N+1) indicates the first vacant position and is equal to the number of nonzero off-diagonal elements +1.

ITAG, LES, DE and V.

Columns in CE should contain nonzero elements ordered according to increasing indices of their row numbers. Because symmetry in structure is assumed, the table ITAG contains the row indices of the elements stored in CE, as well as the column indices of the elements stored in RE.

Indicator IAR may be used when the coefficient matrix does not have structural symmetry or for the purpose of checking the data. If the initial value of IAR is equal to 2 the program will check whether the coefficient matrix is symmetrical. If it is not, zero coefficients are added to make it symmetrical, and the user is informed accordingly by the statement

INITIAL COEFFICIENT MATRIX NONSYMMETRICAL

With IAR = 2 the program will check also whether the elements in each column of CE are stored according to increasing indices of their row numbers. If not the program will reorder them and the user will be notified by the statement

REORDERING OF COLUMNS

In case a row or a column has no off-diagonal terms, i.e., it consists only of a diagonal term, additional elements of zero value should be stored in the CE matrix.

After the main subroutine is executed, the user should check if the indicator for insufficient area IAR is equal to zero. If not the dimension of the arrays ITAG, LNXT, RE and CE should be increased.

IV. GENERAL INFORMATION

Use of COMMON:	None
Workspace:	Provided by the user; see arguments N and LES
Other subroutines:	DATARS, DATBRS, SORDRS, REDURS, SOLVRS
	DATARN, DATBRN, SORDRN, REDURN, SOLVRN
	DATACS, DATBCS, SORDCS, REDUCS, SOLVCS
	DATACN, DATBCN, SORDCN, REDUCN, SOLVCN
	SYMCOLR, SYMCOLC
Input/Output:	None
Restrictions:	$N \geq 1$, $LES \geq LF$
Flags:	IAR

V. EXAMPLES

Example 1

Solve the following system of linear equations:

$$
\begin{aligned}
2x_1 + x_2 &= 0 \\
-0.8x_1 + 2x_2 + x_5 &= 1 \\
0.1x_2 + 10x_3 - 0.5x_5 &= 0 \\
x_3 + 2x_4 + 0.1x_5 &= 0 \\
7.2x_5 &= 1 \\
0.5x_5 + 3x_6 + 0.3x_9 &= 0 \\
-0.3x_4 + 2x_7 + 0.1x_8 &= -2 \\
0.1x_7 + 10x_8 + 0.2x_{10} &= 0 \\
0.3x_4 + 2x_9 &= 0 \\
-x_9 + 3x_{10} &= 3
\end{aligned}
$$

There are no off-diagonal elements in the sixth column of the coefficient matrix, so additional elements of zero value have been added.

Input data for Example 1 is stored in the DATA instructions in the user's program EXAMP1 as shown on page 12. Page 13 shows the results.

```
C
      PROGRAM EXAMP1(OUTPUT,TAPE6=OUTPUT)
C
      INTEGER ITAG(30),LCOL(11),LNXT(30),NOZE(10),NSEQ(10)
      REAL CE(30),V(10),RE(30),DE(10)
C
C         THIS IS THE MAIN PROGRAM FOR SOLVING THE SYSTEM OF LINEAR
C
C         EQUATIONS DESCRIBED IN EXAMPLE 1 USING THE PACKAGE SSLE
C
      DATA LCOL/1,2,4,5,7,11,12,13,14,16,17/
      DATA ITAG/2,1,3,4,7,9,2,3,4,6,9,8,7,6,10,8,14*0/
      DATA CE/-.8,1.,.1,1.,-.3,.3,1.,-.5,.1,.5,0.,.1,.1,.3,-1.,.2,14*0/
      DATA DE/2.,2.,10.,2.,7.2,3.,2.,10.,2.,3./
      DATA V/0.,1.,0.,0.,1.,0.,-2.,0.,0.,3./
      DATA N/10/,LES/30/
      IAR=2
      WRITE(6,60)
      CALL SECOND(TM1)
      CALL SSLERN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR)
      IF(IAR.EQ.1)STOP
      CALL SECOND(TM2)
      CPU=TM2-TM1
      WRITE(6,70)CPU
   70 FORMAT(/," CPU TIME:",F8.3," SECONDS")
   60 FORMAT(1H ,"EXAMPLE 1",/)
      WRITE(6,100)
      IF(N.NE.4.AND.N.NE.2)WRITE(6,120)
      IF(N.EQ.4.OR.N.EQ.2)WRITE(6,130)
      N3=(N+2)/3
      DO 2 I=1,N3
      LI=I
      LO=I+2*N3
      IF(LO.GT.N)LO=I+N3
    2 WRITE(6,110)(J,V(J),J=LI,LO,N3)
  130 FORMAT(1H ," VARIABLE      VALUE        VARIABLE        VALUE",/)
  120 FORMAT(1H ," VARIABLE      VALUE        VARIABLE        VALUE       VARIABL
     +E     VALUE",/)
  110 FORMAT(1X,4(I6,2X,E15.8))
  100 FORMAT(1H ,/," SOLUTION",/)
      STOP
      END
```

EXAMPLE 1

INITIAL COEFFICIENT MATRIX NONSYMMETRICAL
REORDERING OF COLUMNS

CPU TIME: .010 SECONDS

SOLUTION

VARIABLE	VALUE	VARIABLE	VALUE	VARIABLE	VALUE
1	-.17939815E+00	5	.13888889E+00	9	.12934028E-02
2	.35879630E+00	6	-.23277488E-01	10	.10004311E+01
3	.33564815E-02	7	-.10007934E+01		
4	-.86226852E-02	8	-.10000689E-01		

Example 2

Solve the following system of linear complex equations.

$$
\begin{aligned}
x_1 + 2x_2 &= 0 \\
2x_1 + 2x_2 - x_3 &= 1 \\
-(1+j)x_2 + 3x_3 + 4x_4 &= 0 \\
4x_3 + 2x_4 &= 0
\end{aligned}
$$

Input data for Example 2 is stored in the DATA instructions in the user's program EXAMP2 as shown on page 15. Page 16 shows the results.

```
C
      PROGRAM EXAMP2(OUTPUT,TAPE6=OUTPUT)
C
      INTEGER ITAG(30),LCOL(10),LNXT(30),NOZE(10),NSEQ(10)
      COMPLEX CE(30),V(10),RE(30),DE(10)
C
C         THIS IS THE MAIN PROGRAM FOR SOLVING THE SYSTEM OF LINEAR
C
C         EQUATIONS DESCRIBED IN EXAMPLE 2 USING THE PACKAGE SSLE
C
      DATA LCOL/1,2,4,6,7,5*0/
      DATA ITAG/2,1,3,2,4,3,24*0/
      DATA CE/(2.,0.),(2.,0.),(-1.,-1.),(-1.,0.),
     +(4.,0.),(4.,0.),24*((0.,0.))/,V/2*((1.,0.)),8*((0.,0.))/
      DATA DE/(1.,0.),(2.,0.),(3.,0.),(2.,0.),6*((0.,0.))/
      N=4
      LES=30
      WRITE(6,60)
      CALL SECOND(TM1)
      CALL SSLECN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR)
      IF(IAR.EQ.1)STOP
      CALL SECOND(TM2)
      CPU=TM2-TM1
      WRITE(6,70)CPU
   70 FORMAT(/," CPU TIME:",F8.3," SECONDS")
   60 FORMAT(1H ," EXAMPLE 2",/)
      WRITE(6,100)
      IF(N.EQ.4.OR.N.EQ.2)GOTO4
      WRITE(6,120)
      WRITE(6,140)
      GOTO3
    4 WRITE(6,130)
      WRITE(6,150)
    3 N3=(N+2)/3
      DO 2 I=1,N3
      LI=I
      LO=I+2*N3
      IF(LO.GT.N)LO=I+N3
    2 WRITE(6,110)(J,V(J),J=LI,LO,N3)
  140 FORMAT(1H ,"                    REAL           IMAGINARY
     +REAL           IMAGINARY                    REAL           IMAGINARY",/)
  150 FORMAT(1H ,"                    REAL           IMAGINARY
     +REAL           IMAGINARY",/)
  130 FORMAT(1H ," VARIABLE                    VALUE                 VARIABLE
     +       VALUE",/)
  120 FORMAT(1H ," VARIABLE                    VALUE                 VARIABLE
     +       VALUE             VARIABLE                 VALUE",/)
  110 FORMAT(1X,3(I6,2X,2E15.8))
  100 FORMAT(1H ,/," SOLUTION",/)
      STOP
      END
```

EXAMPLE 2

CPU TIME: .003 SECONDS

SOLUTION

VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY
1	-.97560976E-01	-.12195122E+00	3	-.97560976E-01	-.12195122E+00
2	.54878049E+00	.60975610E-01	4	.19512195E+00	.24390244E+00

VI. REFERENCE

[1] K. Zollenkopf, "Bi-factorization - basic computational algorithm and programming techniques", in Large Sparse Sets of Linear Equations, J.K. Reid, Ed., New York: Academic Press, 1971, pp. 75-96.

APPENDIX

LISTING OF THE SSLE PACKAGE

Subroutine	Number of Lines (source text)	Number of Words (compiled code)	Listing from Page
SSLERS	70	233	19
SORDRS	87	236	20
REDURS	67	177	21
SOLVRS	42	103	22
DATARS	31	75	23
DATBRS	27	75	23
SSLERN	77	254	24
SORDRN	94	243	25
REDURN	54	172	26
SOLVRN	44	107	27
DATARN	31	75	28
DATBRN	45	143	28
SYMCOLR	95	207	29
SSLECS	70	233	30
SORDCS	87	237	31
REDUCS	67	224	33
SOLVCS	42	132	34
DATACS	31	76	34
DATBCS	27	75	35
SSLECN	77	254	35
SORDCN	94	274	36
REDUCN	54	235	38
SOLVCN	44	141	39
DATACN	31	76	39
DATBCN	45	154	40
SYMCOLC	94	217	41
Total	1527	5756	

```
C
      SUBROUTINE SSLERS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,V,LF,LES,IAR)
C
C         THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS
C
C         WHOSE COEFFICIENT MATRIX IS REAL AND SYMMETRICAL
C
C         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971.
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      REAL CE(1),V(1)
C
C         N     IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C         LCOL STARTING POSITION OF COLUMNS
C              N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C              POSITION. DIMENSION IS EQUAL TO N+1
C
C         NOZE NUMBER OF NON-ZERO TERMS
C              CALCULATED BY THE DATBRS SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         NSEQ SEQUENCE OF PIVOTAL INDICES
C              CALCULATED BY THE DATARS SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         ITAG ROW INDEX OF ELEMENTS STORED IN CE
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C
C         LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C              CALCULATED BY THE DATARS SUBROUTINE
C
C         CE    IS THE MATRIX OF REAL COEFFICIENTS OF A GIVEN EQUATION
C
C         V    AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C              SIDE VECTOR
C              AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C         LF    INDICATOR FOR THE NEXT VACANT LOCATION
C              CALCULATED BY THE DATARS SUBROUTINE
C
C         LES  NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE 8 CE
C
C         IAR   INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C              ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C              AREA
C
      CALL DATARS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
      CALL DATBRS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
      IAR=0
      CALL SORDRS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL REDURS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR)
```

```
      IF (IAR.EQ.1) RETURN
      CALL SOLVRS (N,LCOL,NSEQ,ITAG,LNXT,CE,V)
      RETURN
      END
C
C
      SUBROUTINE SORDRS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
C
C         THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
C
C         FOR SYMMETRICAL MATRICES
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1)
      N1=N-1
      DO 140 J=1,N1
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      J1=J+1
C
C          FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS
C
      DO 10 I=J1,N
      K=NSEQ(I)
      IF (NOZE(K).GE.MIN) GO TO 10
      MIN=NOZE(K)
      M=I
   10 CONTINUE
C
C          INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M
C
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
C
C          TAKE THE SUCCESSFUL ELEMENT FROM MINIMAL COLUMN M
C          IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE
C
   20 K=ITAG(LK)
      IF (K.EQ.KP) GO TO 100
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
   30 IF (I-IP) 40,50,80
   40 LA=L
      L=LNXT(L)
      IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 30
   50 IF (I.NE.KP) GO TO 60
      LN=LNXT(L)
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LNXT(L)=LF
      LF=L
      CE(L)=0.
      NOZE(K)=NOZE(K)-1
      L=LN
      GO TO 70
   60 LA=L
      L=LNXT(L)
```

```
   70 IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 90
   80 IF (LF.LE.0) GO TO 110
      LN=LF
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LF=LNXT(LN)
      LNXT(LN)=L
      ITAG(LN)=IP
      NOZE(K)=NOZE(K)+1
      LA=LN
   90 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 100
      IP=ITAG(LI)
      GO TO 30
  100 LK=LNXT(LK)
      GO TO 130
  110 WRITE (6,120)
      IAR=1
  120 FORMAT (" DIMENSIONED AREA TO SMALL IN SORDRS")
      RETURN
  130 IF (LK.GT.0) GO TO 20
  140 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE REDURS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR)
C
C         THIS SUBROUTINE PERFORMS REDUCTION OF A SYMMETRICAL MATRIX
C
C         THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),D,CF
      DO 110 J=1,N
      KP=NSEQ(J)
      LK=LCOL(KP)
      LP=LF
   10 IF (LP.LE.0) GO TO 120
      K=ITAG(LK)
      IF (K.NE.KP) GO TO 20
C
C         DIVISION BY DIAGONAL ELEMENT
C
      D=1./CE(LK)
      CE(LK)=D
      GO TO 30
   20 CE(LP)=CE(LK)
   30 LK=LNXT(LK)
      IF (LK.LE.0) GO TO 40
      LP=LNXT(LP)
      GO TO 10
   40 LK=LCOL(KP)
   50 K=ITAG(LK)
      IF (K.EQ.KP) GO TO 100
C
C         ELEMENTS OF FACTOR MATRICES L(J) AND R(J)
C
      CF=D*CE(LK)
      CE(LK)=-CF
      LP=LF
      LI=LCOL(KP)
      IP=ITAG(LI)
```

```
      L=LCOL(K)
      I=ITAG(L)
   60 IF (I-IP) 70,80,90
   70 L=LNXT(L)
      IF (L.LE.0) GO TO 100
      I=ITAG(L)
      GO TO 60
C
C          ELEMENTS OF REDUCED MATRIX A(J)
C
   80 CE(L)=CE(L)-CF*CE(LP)
      L=LNXT(L)
      IF (L.LE.0) GO TO 100
      I=ITAG(L)
   90 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 100
      IP=ITAG(LI)
      LP=LNXT(LP)
      GO TO 60
  100 LK=LNXT(LK)
      IF (LK.GT.0) GO TO 50
  110 CONTINUE
      RETURN
  120 WRITE (6,130)
      IAR=1
  130 FORMAT (" DIMENSIONED AREA TOO SMALL IN REDURS")
      RETURN
      END
C
C
      SUBROUTINE SOLVRS (N,LCOL,NSEQ,ITAG,LNXT,CE,V)
C
C          THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF
C
C          MATRIX MULTIPLICATIONS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),V(1),CF,SUM
C
C          FIRST PART OF THE SOLUTION
C          B1=L(N)*L(N-1)*...*L(1)*B
C
      DO 20 J=1,N
      K=NSEQ(J)
      CF=V(K)
      V(K)=0.
      L=LCOL(K)
   10 I=ITAG(L)
      V(I)=V(I)+CE(L)*CF
      L=LNXT(L)
      IF (L.GT.0) GO TO 10
   20 CONTINUE
C
C          SECOND PART OF THE SOLUTION
C          X=R(1)*R(2)*...*R(N)*B1
C
      N1=N-1
      DO 40 JJ=1,N1
      J=N-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 I=ITAG(L)
      IF (I.NE.K) SUM=SUM+CE(L)*V(I)
      L=LNXT(L)
```

```
      IF (L.GT.0) GO TO 30
      V(K)=SUM
   40 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DATARS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NSEQ,LNXT,LF
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1)
      LF=LCOL(N+1)
      DO 10 I=1,LES
      LNXT(I)=I+1
   10 CONTINUE
      LNXT(LES)=0
      DO 20 I=LF,LES
      CE(I)=0.
   20 CONTINUE
C
C         CALCULATE MATRICES NSEQ 8 LNXT
C
      DO 50 I=1,N
      J=I+1
      IF (LCOL(I).EQ.0) GO TO 50
   30 IF (LCOL(J).NE.0) GO TO 40
      J=J+1
      GO TO 30
   40 LNXT(LCOL(I+1)-1)=0
   50 NSEQ(I)=I
      RETURN
      END
C
C
      SUBROUTINE DATBRS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NOZE
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1)
C
C         IF IAR.EQ.2 CHECK SYMMETRY
C
      IF (IAR.EQ.2) CALL SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         CALCULATE MATRIX NOZE
C
      DO 20 I=1,N
      NOZE(I)=0
      NE=LCOL(I)
   10 IF (NE.EQ.0) GO TO 20
      NOZE(I)=NOZE(I)+1
      NE=LNXT(NE)
      GO TO 10
   20 CONTINUE
      RETURN
      END
C
C
```

```
      SUBROUTINE SSLERN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IA
     1R)
C
C         THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS
C
C         WHOSE COEFFICIENT MATRIX IS REAL AND NONSYMMETRICAL
C
C         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971.
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      REAL CE(1),V(1),RE(1),DE(1)
C
C        N    IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C        LCOL STARTING POSITION OF COLUMNS
C             N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C             POSITION. DIMENSION IS EQUAL TO N+1
C
C        NOZE NUMBER OF NON-ZERO TERMS
C             CALCULATED BY THE DATBRN SUBROUTINE
C             DIMENSION IS EQUAL TO N
C
C        NSEQ SEQUENCE OF PIVOTAL INDICES
C             CALCULATED BY THE DATARN SUBROUTINE
C             DIMENSION IS EQUAL TO N
C
C        ITAG ROW INDEX OF ELEMENTS STORED IN CE
C             DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C             ELEMENTS
C
C        LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C             DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C             ELEMENTS
C             CALCULATED BY THE DATARN SUBROUTINE
C
C        CE   IS THE MATRIX OF REAL COEFFICIENTS OF A GIVEN EQUATION
C
C        RE   AN ARRAY HAVING ROWWISE STORED TERMS OF THE COEFFICIENT
C             MATRIX. CALCULATED BY THE DATBRN SUBROUTINE
C
C        DE   AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF CO-
C             EFFICIENT MATRIX. DATA REQUIRED FROM THE USER
C
C        V    AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C             SIDE VECTOR
C             AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C        LF   INDICATOR FOR THE NEXT VACANT LOCATION
C             CALCULATED BY THE DATARN SUBROUTINE
C
C        LES  NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C        IAR  INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C             ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C             AREA
C
      CALL DATARN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
```

```
C
      CALL DATBRN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      CALL SORDRN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL REDURN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL SOLVRN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      RETURN
      END
C
C
      SUBROUTINE SORDRN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR)
C
C         THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
C
C         FOR NONSYMMETRICAL MATRICES
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),RE(1)
      N1=N-1
      DO 140 J=1,N1
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      J1=J+1
C
C         FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS
C
      DO 10 I=J1,N
      K=NSEQ(I)
      IF (NOZE(K).GE.MIN) GO TO 10
      MIN=NOZE(K)
      M=I
   10 CONTINUE
C
C         INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M
C
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
   20 IF (LK.LE.0) GO TO 140
C
C         TAKE THE NEXT ELEMENT FROM MINIMAL COLUMN M
C         IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE
C
      K=ITAG(LK)
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
   30 IF (I-IP) 50,40,90
   40 LA=L
      L=LNXT(L)
      IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 100
   50 IF (I.NE.KP) GO TO 60
      LN=LNXT(L)
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LNXT(L)=LF
      LF=L
```

```
      CE(L)=0.
      RE(L)=0.
      NOZE(K)=NOZE(K)-1
      L=LN
      GO TO 70
   60 LA=L
      L=LNXT(L)
   70 IF (L.LE.0) GO TO 80
      I=ITAG(L)
      GO TO 30
   80 IF (LI.LE.0) GO TO 120
      I=N+1
      GO TO 30
   90 IF (IP.EQ.K) GO TO 100
      IF (LF.LE.0) GO TO 130
      LN=LF
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LF=LNXT(LN)
      LNXT(LN)=L
      ITAG(LN)=IP
      NOZE(K)=NOZE(K)+1
      LA=LN
  100 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 110
      IP=ITAG(LI)
      GO TO 30
  110 IF (L.GT.0) IP=N+1
      IF (L.GT.0) GO TO 30
  120 LK=LNXT(LK)
      GO TO 20
  130 WRITE (6,150)
      IAR=1
      RETURN
  140 CONTINUE
      RETURN
C
  150 FORMAT (" DIMENSIONED AREA TO SMALL IN SORDRN")
      END
C
C
      SUBROUTINE REDURN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,LF,IAR)
C
C         THIS SUBROUTINE PERFORMS REDUCTION OF A NONSYMMETRICAL MATRIX
C
C         THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),RE(1),DE(1),D,CF,RF
      DO 100 J=1,N
      KP=NSEQ(J)
      D=1/DE(KP)
      DE(KP)=D
      LK=LCOL(KP)
      IF (LK.LE.0) GO TO 100
   10 RE(LK)=D*RE(LK)
      LK=LNXT(LK)
      IF (LK.GT.0) GO TO 10
      LK=LCOL(KP)
   20 K=ITAG(LK)
C
C         ELEMENTS OF FACTOR MATRICES L(J) AND R(J)
C
      CF=RE(LK)
      RF=CE(LK)
```

```
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
   30 IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
   40 IF (I-IP) 50,60,70
   50 L=LNXT(L)
      GO TO 30
C
C         ELEMENTS OF REDUCED MATRIX A(J)
C
   60 CE(L)=CE(L)-CF*CE(LI)
      RE(L)=RE(L)-RF*RE(LI)
      L=LNXT(L)
      IF (L.LE.0) I=N+1
      IF (L.GT.0) I=ITAG(L)
      GO TO 80
   70 IF (IP.NE.K) GO TO 80
      DE(K)=DE(K)-CF*CE(LI)
   80 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 90
      IP=ITAG(LI)
      GO TO 40
   90 LK=LNXT(LK)
      IF (LK.GT.0) GO TO 20
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOLVRN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
C
C         THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF
C
C         MATRIX MULTIPLICATIONS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),RE(1),DE(1),V(1),CF,SUM
C
C         FIRST PART OF THE SOLUTION
C         B1=L(N)*L(N-1)*...*L(1)*B
C
      DO 20 J=1,N
      K=NSEQ(J)
      CF=DE(K)*V(K)
      V(K)=CF
      L=LCOL(K)
   10 IF (L.LE.0) GO TO 20
      I=ITAG(L)
      V(I)=V(I)-CE(L)*CF
      L=LNXT(L)
      GO TO 10
   20 CONTINUE
C
C         SECOND PART OF THE SOLUTION
C         X=R(1)*R(2)*...*R(N)*B1
C
      N1=N-1
      DO 50 JJ=1,N1
      J=N-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 IF (L.LE.0) GO TO 40
      I=ITAG(L)
```

```
      SUM=SUM-RE(L)*V(I)
      L=LNXT(L)
      GO TO 30
   40 V(K)=SUM
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DATARN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NSEQ,LNXT,LF
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),RE(1)
      LF=LCOL(N+1)
      DO 10 I=1,LES
      LNXT(I)=I+1
   10 CONTINUE
      LNXT(LES)=0
      DO 20 I=LF,LES
      CE(I)=0.
   20 CONTINUE
C
C         CALCULATE MATRICES NSEQ 8 LNXT
C
      DO 50 I=1,N
      J=I+1
      IF (LCOL(I).EQ.0) GO TO 50
   30 IF (LCOL(J).NE.0) GO TO 40
      J=J+1
      GO TO 30
   40 LNXT(LCOL(I+1)-1)=0
   50 NSEQ(I)=I
      RETURN
      END
C
C
      SUBROUTINE DATBRN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NOZE,RE
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1),RE(1)
C
C         IF IAR.EQ.2 CHECK SYMMETRY
C
      IF (IAR.EQ.2) CALL SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         CALCULATE MATRIX NOZE
C
      DO 20 I=1,N
      NOZE(I)=0
      NE=LCOL(I)
   10 IF (NE.EQ.0) GO TO 20
      NOZE(I)=NOZE(I)+1
      NE=LNXT(NE)
      GO TO 10
   20 CONTINUE
      DO 30 I=1,LES
      RE(I)=0.
```

```
   30 CONTINUE
C
C         CALCULATE ROWWISE STORED COEFFICIENT MATRIX
C
      DO 70 I=1,N
      NE=LCOL(I)
   40 IF (NE.EQ.0) GO TO 70
      NR=ITAG(NE)
      LI=LCOL(NR)
   50 IF (ITAG(LI).EQ.I) GO TO 60
      LI=LNXT(LI)
      GO TO 50
   60 RE(LI)=CE(NE)
      NE=LNXT(NE)
      GO TO 40
   70 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         THIS SUBROUTINE CHECKS THE STRUCTURAL SYMMETRY AND ORDERING OF
C
C         COLUMNS OF THE COEFFICIENT MATRIX. IF THE MATRIX IS NOT
C
C         SYMMETRICAL, NEW ZERO COEFFICIENTS ARE ADDED TO REACH SYMMETRY
C
C         COLUMNS ARE REORDERED IF NECESSARY
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      REAL CE(1)
      IS=0
      DO 70 I=1,N
      LC=LCOL(I)
      IF (LC.EQ.0) GO TO 70
   10 IR=ITAG(LC)
C
C         CHECK IF THERE EXISTS AN ELEMENT SYMMETRICAL TO THE ONE
C         CONSIDERED
C
      NLC=LCOL(IR)
   20 IF (NLC.EQ.0) GO TO 30
      LLC=NLC
      LLR=ITAG(LLC)
      IF (LLR.EQ.I) GO TO 60
      NLC=LNXT(LLC)
      GO TO 20
C
C         MATRIX IS NONSYMMETRICAL-ADD NECESSARY ZERO COEFFICIENT
C
   30 IF (LCOL(IR).NE.0) GO TO 40
      LCOL(IR)=LF
      GO TO 50
   40 LNXT(LLC)=LF
   50 CE(LF)=0.
      IS=1
      ITAG(LF)=I
      NOZE(IR)=NOZE(IR)+1
      IF=LF
      LF=LNXT(IF)
      LNXT(IF)=0
C
C         PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C         CONSIDERATION
```

```
C
   60 LC=LNXT(LC)
      IF (LC.NE.0) GO TO 10
   70 CONTINUE
      IF (IS.EQ.1) WRITE (6,130)
C
C          ROWS AND COLUMNS ARE ORDERED SUCH THAT INCREASING VALUES OF
C          INDICES DESCRIBE THE NEXT NONZERO ELEMENTS IN EACH ROW AND
C          COLUMN
C
      IS=0
      DO 120 I=1,N
      I1=LCOL(I)
   80 II=I1
   90 I2=LNXT(II)
      IF (I2.EQ.0) GO TO 110
      IF (ITAG(I1).LT.ITAG(I2)) II=I2
      IF (ITAG(I1).LT.ITAG(I2)) GO TO 90
      IS=1
      IF (I1.EQ.LCOL(I)) GO TO 100
C
C          EXCHANGE TWO ELEMENTS
C
      LNXT(I0)=I2
      LNXT(II)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
  100 LCOL(I)=I2
      LNXT(II)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
C
C          PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C          CONSIDERATION
C
  110 I0=I1
      I1=LNXT(I1)
      IF (I1.EQ.0) GO TO 120
      GO TO 80
  120 CONTINUE
      IF (IS.EQ.1) WRITE (6,140)
      RETURN
C
  130 FORMAT (1H ," INITIAL COEFFICIENT MATRIX NONSYMMETRICAL")
  140 FORMAT (1H ," REORDERING OF COLUMNS")
      END
C
C
      SUBROUTINE SSLECS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,V,LF,LES,IAR)
C
C          THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS
C
C          WHOSE COEFFICIENT MATRIX IS COMPLEX AND SYMMETRICAL
C
C          BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C          " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C          - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C          MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C          J.K.REID, EDITOR, ACADEMIC PRESS, 1971.
```

```
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      COMPLEX CE(1),V(1)
C
C         N     IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C         LCOL STARTING POSITION OF COLUMNS
C              N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C              POSITION. DIMENSION IS EQUAL TO N+1
C
C         NOZE NUMBER OF NON-ZERO TERMS
C              CALCULATED BY THE DATBCS SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         NSEQ SEQUENCE OF PIVOTAL INDICES
C              CALCULATED BY THE DATACS SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         ITAG ROW INDEX OF ELEMENTS STORED IN CE
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C
C         LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C              CALCULATED BY THE DATACS SUBROUTINE
C
C         CE    IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
C
C         V     AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C               SIDE VECTOR
C               AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C         LF    INDICATOR FOR THE NEXT VACANT LOCATION
C               CALCULATED BY THE DATACS SUBROUTINE
C
C         LES   NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C         IAR   INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C               ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C               AREA
C
      CALL DATACS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
      CALL DATBCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
      IAR=0
      CALL SORDCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL REDUCS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL SOLVCS (N,LCOL,NSEQ,ITAG,LNXT,CE,V)
      RETURN
      END
C
C
      SUBROUTINE SORDCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
C
C         THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
C
C         FOR SYMMETRICAL MATRICES
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1)
      N1=N-1
      DO 140 J=1,N1
```

```
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      J1=J+1
C
C          FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS
C
      DO 10 I=J1,N
      K=NSEQ(I)
      IF (NOZE(K).GE.MIN) GO TO 10
      MIN=NOZE(K)
      M=I
   10 CONTINUE
C
C          INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M
C
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
C
C          TAKE THE SUCCESSFUL ELEMENT FROM MINIMAL COLUMN M
C          IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE
C
   20 K=ITAG(LK)
      IF (K.EQ.KP) GO TO 100
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
   30 IF (I-IP) 40,50,80
   40 LA=L
      L=LNXT(L)
      IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 30
   50 IF (I.NE.KP) GO TO 60
      LN=LNXT(L)
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LNXT(L)=LF
      LF=L
      CE(L)=CMPLX(0.,0.)
      NOZE(K)=NOZE(K)-1
      L=LN
      GO TO 70
   60 LA=L
      L=LNXT(L)
   70 IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 90
   80 IF (LF.LE.0) GO TO 110
      LN=LF
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LF=LNXT(LN)
      LNXT(LN)=L
      ITAG(LN)=IP
      NOZE(K)=NOZE(K)+1
      LA=LN
   90 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 100
      IP=ITAG(LI)
      GO TO 30
```

```
  100 LK=LNXT(LK)
      GO TO 130
  110 WRITE (6,120)
      IAR=1
  120 FORMAT (" DIMENSIONED AREA TO SMALL IN SORDCS")
      RETURN
  130 IF (LK.GT.0) GO TO 20
  140 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE REDUCS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR)
C
C         THIS SUBROUTINE PERFORMS REDUCTION OF A SYMMETRICAL MATRIX
C
C         THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),D,CF
      DO 110 J=1,N
      KP=NSEQ(J)
      LK=LCOL(KP)
      LP=LF
   10 IF (LP.LE.0) GO TO 120
      K=ITAG(LK)
      IF (K.NE.KP) GO TO 20
C
C         DIVISION BY DIAGONAL ELEMENT
C
      D=1./CE(LK)
      CE(LK)=D
      GO TO 30
   20 CE(LP)=CE(LK)
   30 LK=LNXT(LK)
      IF (LK.LE.0) GO TO 40
      LP=LNXT(LP)
      GO TO 10
   40 LK=LCOL(KP)
   50 K=ITAG(LK)
      IF (K.EQ.KP) GO TO 100
C
C         ELEMENTS OF FACTOR MATRICES L(J) AND R(J)
C
      CF=D*CE(LK)
      CE(LK)=-CF
      LP=LF
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
   60 IF (I-IP) 70,80,90
   70 L=LNXT(L)
      IF (L.LE.0) GO TO 100
      I=ITAG(L)
      GO TO 60
C
C         ELEMENTS OF REDUCED MATRIX A(J)
C
   80 CE(L)=CE(L)-CF*CE(LP)
      L=LNXT(L)
      IF (L.LE.0) GO TO 100
      I=ITAG(L)
   90 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 100
```

```
      IP=ITAG(LI)
      LP=LNXT(LP)
      GO TO 60
  100 LK=LNXT(LK)
      IF (LK.GT.0) GO TO 50
  110 CONTINUE
      RETURN
  120 WRITE (6,130)
      IAR=1
  130 FORMAT (" DIMENSIONED AREA TOO SMALL IN REDUCS")
      RETURN
      END
C
C
      SUBROUTINE SOLVCS (N,LCOL,NSEQ,ITAG,LNXT,CE,V)
C
C         THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF
C
C         MATRIX MULTIPLICATIONS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),V(1),CF,SUM
C
C         FIRST PART OF THE SOLUTION
C         B1=L(N)*L(N-1)*...*L(1)*B
C
      DO 20 J=1,N
      K=NSEQ(J)
      CF=V(K)
      V(K)=CMPLX(0.,0.)
      L=LCOL(K)
   10 I=ITAG(L)
      V(I)=V(I)+CE(L)*CF
      L=LNXT(L)
      IF (L.GT.0) GO TO 10
   20 CONTINUE
C
C         SECOND PART OF THE SOLUTION
C         X=R(1)*R(2)*...*R(N)*B1
C
      N1=N-1
      DO 40 JJ=1,N1
      J=N-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 I=ITAG(L)
      IF (I.NE.K) SUM=SUM+CE(L)*V(I)
      L=LNXT(L)
      IF (L.GT.0) GO TO 30
      V(K)=SUM
   40 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DATACS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NSEQ,LNXT,LF
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1)
      LF=LCOL(N+1)
```

```
      DO 10 I=1,LES
      LNXT(I)=I+1
   10 CONTINUE
      LNXT(LES)=0
      DO 20 I=LF,LES
      CE(I)=CMPLX(0.,0.)
   20 CONTINUE
C
C         CALCULATE MATRICES NSEQ & LNXT
C
      DO 50 I=1,N
      J=I+1
      IF (LCOL(I).EQ.0) GO TO 50
   30 IF (LCOL(J).NE.0) GO TO 40
      J=J+1
      GO TO 30
   40 LNXT(LCOL(I+1)-1)=0
   50 NSEQ(I)=I
      RETURN
      END
C
C
      SUBROUTINE DATBCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NOZE
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1)
C
C         IF IAR.EQ.2 CHECK SYMMETRY
C
      IF (IAR.EQ.2) CALL SYMCOLC (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         CALCULATE MATRIX NOZE
C
      DO 20 I=1,N
      NOZE(I)=0
      NE=LCOL(I)
   10 IF (NE.EQ.0) GO TO 20
      NOZE(I)=NOZE(I)+1
      NE=LNXT(NE)
      GO TO 10
   20 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SSLECN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IA
     1R)
C
C         THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS
C
C         WHOSE COEFFICIENT MATRIX IS SPARSE AND NONSYMMETRICAL
C
C         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971.
```

```
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      COMPLEX CE(1),V(1),RE(1),DE(1)
C
C         N      IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C         LCOL STARTING POSITION OF COLUMNS
C              N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C              POSITION. DIMENSION IS EQUAL TO N+1
C
C         NOZE NUMBER OF NON-ZERO TERMS
C              CALCULATED BY THE DATBCN SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         NSEQ SEQUENCE OF PIVOTAL INDICES
C              CALCULATED BY THE DATACN SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         ITAG ROW INDEX OF ELEMENTS STORED IN CE
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C
C         LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C              CALCULATED BY THE DATACN SUBROUTINE
C
C         CE     IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
C
C         RE     AN ARRAY HAVING ROWWISE STORED TERMS OF THE COEFFICIENT
C              MATRIX. CALCULATED BY THE DATBCN SUBROUTINE
C
C         DE     AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF CO-
C              EFFICIENT MATRIX. DATA REQUIRED FROM THE USER
C
C         V      AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C              SIDE VECTOR
C              AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C         LF     INDICATOR FOR THE NEXT VACANT LOCATION
C              CALCULATED BY THE DATACN SUBROUTINE
C
C         LES    NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C         IAR    INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C              ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C              AREA
C
      CALL DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
      CALL DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      CALL SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,LF,IAR)
      IF (IAR.EQ.1) RETURN
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      RETURN
      END
C
C
      SUBROUTINE SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR)
C
C         THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
C
```

```
C          FOR NONSYMMETRICAL MATRICES
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1)
      N1=N-1
      DO 140 J=1,N1
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      J1=J+1
C
C          FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS
C
      DO 10 I=J1,N
      K=NSEQ(I)
      IF (NOZE(K).GE.MIN) GO TO 10
      MIN=NOZE(K)
      M=I
   10 CONTINUE
C
C          INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M
C
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
   20 IF (LK.LE.0) GO TO 140
C
C          TAKE THE NEXT ELEMENT FROM MINIMAL COLUMN M
C          IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE
C
      K=ITAG(LK)
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
   30 IF (I-IP) 50,40,90
   40 LA=L
      L=LNXT(L)
      IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 100
   50 IF (I.NE.KP) GO TO 60
      LN=LNXT(L)
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LNXT(L)=LF
      LF=L
      CE(L)=CMPLX(0.,0.)
      RE(L)=CMPLX(0.,0.)
      NOZE(K)=NOZE(K)-1
      L=LN
      GO TO 70
   60 LA=L
      L=LNXT(L)
   70 IF (L.LE.0) GO TO 80
      I=ITAG(L)
      GO TO 30
   80 IF (LI.LE.0) GO TO 120
      I=N+1
      GO TO 30
   90 IF (IP.EQ.K) GO TO 100
      IF (LF.LE.0) GO TO 130
      LN=LF
```

```
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LF=LNXT(LN)
      LNXT(LN)=L
      ITAG(LN)=IP
      NOZE(K)=NOZE(K)+1
      LA=LN
  100 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 110
      IP=ITAG(LI)
      GO TO 30
  110 IF (L.GT.0) IP=N+1
      IF (L.GT.0) GO TO 30
  120 LK=LNXT(LK)
      GO TO 20
  130 WRITE (6,150)
      IAR=1
      RETURN
  140 CONTINUE
      RETURN
C
  150 FORMAT (" DIMENSIONED AREA TO SMALL IN SORDCN")
      END
C
C
      SUBROUTINE REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,LF,IAR)
C
C         THIS SUBROUTINE PERFORMS REDUCTION OF A NONSYMMETRICAL MATRIX
C
C         THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),D,CF,RF
      DO 100 J=1,N
      KP=NSEQ(J)
      D=1/DE(KP)
      DE(KP)=D
      LK=LCOL(KP)
      IF (LK.LE.0) GO TO 100
   10 RE(LK)=D*RE(LK)
      LK=LNXT(LK)
      IF (LK.GT.0) GO TO 10
      LK=LCOL(KP)
   20 K=ITAG(LK)
C
C         ELEMENTS OF FACTOR MATRICES L(J) AND R(J)
C
      CF=RE(LK)
      RF=CE(LK)
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
   30 IF (L.GT.0) I=ITAG(L)
      IF (L.LE.0) I=N+1
   40 IF (I-IP) 50,60,70
   50 L=LNXT(L)
      GO TO 30
C
C         ELEMENTS OF REDUCED MATRIX A(J)
C
   60 CE(L)=CE(L)-CF*CE(LI)
      RE(L)=RE(L)-RF*RE(LI)
      L=LNXT(L)
      IF (L.LE.0) I=N+1
      IF (L.GT.0) I=ITAG(L)
```

```
      GO TO 80
   70 IF (IP.NE.K) GO TO 80
      DE(K)=DE(K)-CF*CE(LI)
   80 LI=LNXT(LI)
      IF (LI.LE.0) GO TO 90
      IP=ITAG(LI)
      GO TO 40
   90 LK=LNXT(LK)
      IF (LK.GT.0) GO TO 20
  100 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
C
C         THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF
C
C         MATRIX MULTIPLICATIONS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),V(1),CF,SUM
C
C         FIRST PART OF THE SOLUTION
C         B1=L(N)*L(N-1)*...*L(1)*B
C
      DO 20 J=1,N
      K=NSEQ(J)
      CF=DE(K)*V(K)
      V(K)=CF
      L=LCOL(K)
   10 IF (L.LE.0) GO TO 20
      I=ITAG(L)
      V(I)=V(I)-CE(L)*CF
      L=LNXT(L)
      GO TO 10
   20 CONTINUE
C
C         SECOND PART OF THE SOLUTION
C         X=R(1)*R(2)*...*R(N)*B1
C
      N1=N-1
      DO 50 JJ=1,N1
      J=N-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 IF (L.LE.0) GO TO 40
      I=ITAG(L)
      SUM=SUM-RE(L)*V(I)
      L=LNXT(L)
      GO TO 30
   40 V(K)=SUM
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NSEQ,LNXT,LF
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
```

```
      COMPLEX CE(1),RE(1)
      LF=LCOL(N+1)
      DO 10 I=1,LES
      LNXT(I)=I+1
   10 CONTINUE
      LNXT(LES)=0
      DO 20 I=LF,LES
      CE(I)=CMPLX(0.,0.)
   20 CONTINUE
C
C         CALCULATE MATRICES NSEQ & LNXT
C
      DO 50 I=1,N
      J=I+1
      IF (LCOL(I).EQ.0) GO TO 50
   30 IF (LCOL(J).NE.0) GO TO 40
      J=J+1
      GO TO 30
   40 LNXT(LCOL(I+1)-1)=0
   50 NSEQ(I)=I
      RETURN
      END
C
C
      SUBROUTINE DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NOZE,RE
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1)
C
C         IF IAR.EQ.2 CHECK SYMMETRY
C
      IF (IAR.EQ.2) CALL SYMCOLC (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         CALCULATE MATRIX NOZE
C
      DO 20 I=1,N
      NOZE(I)=0
      NE=LCOL(I)
   10 IF (NE.EQ.0) GO TO 20
      NOZE(I)=NOZE(I)+1
      NE=LNXT(NE)
      GO TO 10
   20 CONTINUE
      DO 30 I=1,LES
      RE(I)=CMPLX(0.,0.)
   30 CONTINUE
C
C         CALCULATE ROWWISE STORED COEFFICIENT MATRIX
C
      DO 70 I=1,N
      NE=LCOL(I)
   40 IF (NE.EQ.0) GO TO 70
      NR=ITAG(NE)
      LI=LCOL(NR)
   50 IF (ITAG(LI).EQ.I) GO TO 60
      LI=LNXT(LI)
      GO TO 50
   60 RE(LI)=CE(NE)
      NE=LNXT(NE)
      GO TO 40
   70 CONTINUE
```

```
      RETURN
      END
C
C
      SUBROUTINE SYMCOLC (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         THIS SUBROUTINE CHECKS THE STRUCTURAL SYMMETRY AND ORDERING OF
C
C         COLUMNS OF THE COEFFICIENT MATRIX. IF THE MATRIX IS NOT
C
C         SYMMETRICAL, NEW ZERO COEFFICIENTS ARE ADDED TO REACH SYMMETRY
C
C         COLUMNS ARE REORDERED IF NECESSARY
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1)
      IS=0
      DO 70 I=1,N
      LC=LCOL(I)
      IF (LC.EQ.0) GO TO 70
   10 IR=ITAG(LC)
C
C         CHECK IF THERE EXISTS AN ELEMENT SYMMETRICAL TO THE ONE
C         CONSIDERED
C
      NLC=LCOL(IR)
   20 IF (NLC.EQ.0) GO TO 30
      LLC=NLC
      LLR=ITAG(LLC)
      IF (LLR.EQ.I) GO TO 60
      NLC=LNXT(LLC)
      GO TO 20
C
C         MATRIX IS NONSYMMETRICAL-ADD NECESSARY ZERO COEFFICIENT
C
   30 IF (LCOL(IR).NE.0) GO TO 40
      LCOL(IR)=LF
      GO TO 50
   40 LNXT(LLC)=LF
   50 CE(LF)=CMPLX(0.,0.)
      IS=1
      ITAG(LF)=I
      NOZE(IR)=NOZE(IR)+1
      IF=LF
      LF=LNXT(IF)
      LNXT(IF)=0
C
C         PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C         CONSIDERATION
C
   60 LC=LNXT(LC)
      IF (LC.NE.0) GO TO 10
   70 CONTINUE
      IF (IS.EQ.1) WRITE (6,130)
C
C         ROWS AND COLUMNS ARE ORDERED SUCH THAT INCREASING VALUES OF
C         INDICES DESCRIBE THE NEXT NONZERO ELEMENTS IN EACH ROW AND
C         COLUMN
C
      IS=0
      DO 120 I=1,N
      I1=LCOL(I)
   80 II=I1
   90 I2=LNXT(II)
      IF (I2.EQ.0) GO TO 110
```

```
      IF (ITAG(I1).LT.ITAG(I2)) II=I2
      IF (ITAG(I1).LT.ITAG(I2)) GO TO 90
      IS=1
      IF (I1.EQ.LCOL(I)) GO TO 100
C
C         EXCHANGE TWO ELEMENTS
C
      LNXT(I0)=I2
      LNXT(I1)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
  100 LCOL(I)=I2
      LNXT(I1)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
C
C         PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C         CONSIDERATION
C
  110 I0=I1
      I1=LNXT(I1)
      IF (I1.EQ.0) GO TO 120
      GO TO 80
  120 CONTINUE
      IF (IS.EQ.1) WRITE (6,140)
      RETURN
C
  130 FORMAT (1H ," INITIAL COEFFICIENT MATRIX NONSYMMETRICAL")
  140 FORMAT (1H ," REORDERING OF COLUMNS")
```

SOC-289

SSLE - A FORTRAN PACKAGE FOR SOLVING SPARSE LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

April 1982 No. of pages: 42

Revised:

Key Words: Linear systems, sparse matrices, solution of real or complex linear equations

Abstract: SSLE is a package of subroutines for solving systems of sparse linear equations. The bi-factorization method is used for systems having a symmetrical structure of the coefficient matrix. The method is based on Zollenkopf's algorithm. In the case of a non-symmetrical structure necessary zero elements are added. There are four versions of the principal subroutine appropriate for a symmetrical or nonsymmetrical as well as for a real or complex coefficient matrix.

Description: Contains Fortran listing, user's manual. Source deck or magnetic tape available for $100.00. The listing contains 1527 lines, of which 595 are comments.

Related Work:

Price: $100.00.