

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	DATA CN
B	DATBRS	DATBRN	DATBCS	DATBCN
C	SORDRS	SORDRN	SORDCS	SORDCN
D	REDURS	REDURN	REDUCS	REDUCN
E	SOLVRS	SOLVRN	SOLVCS	SOLVCN
F	SYMCOLR		SYMCOLC	

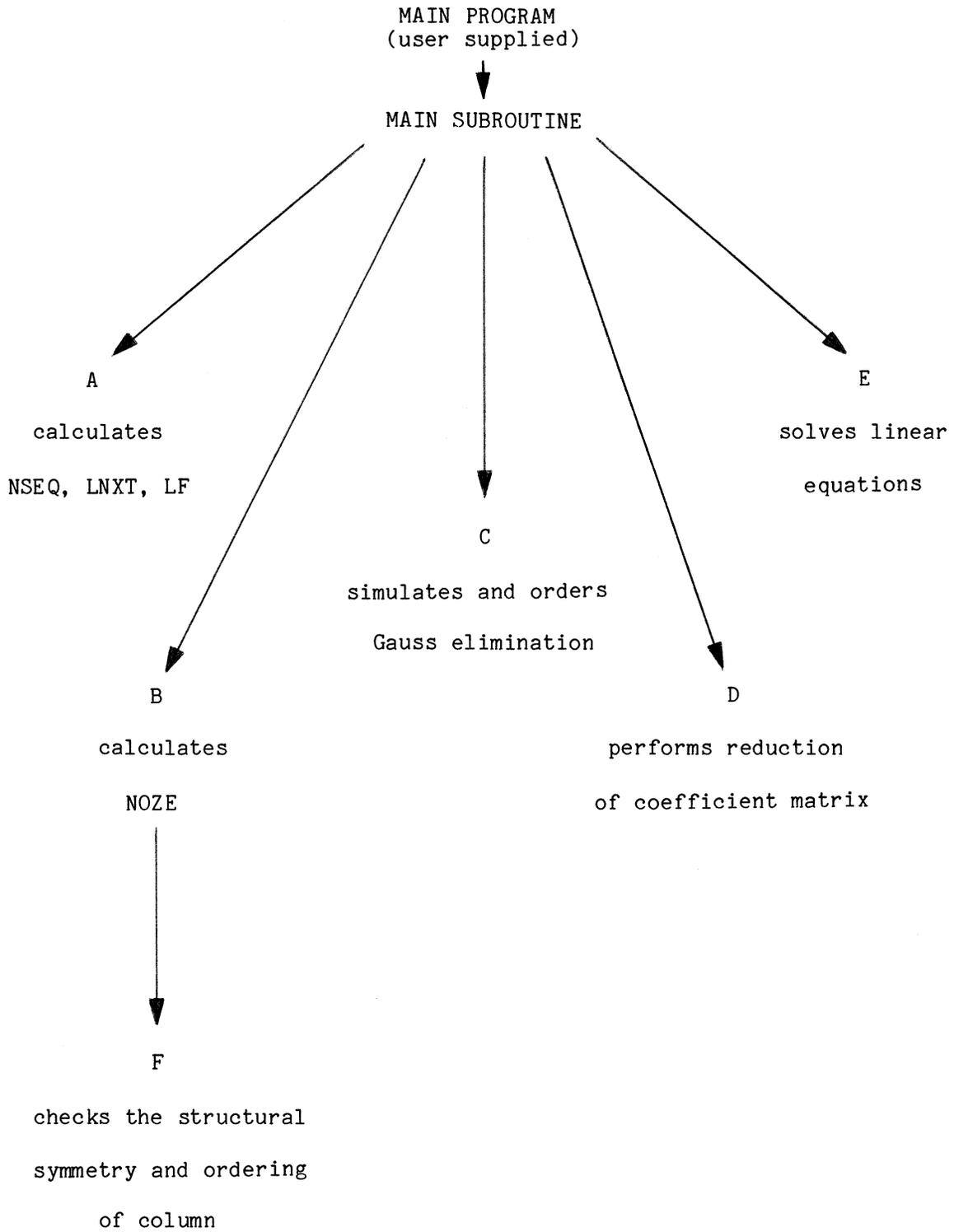


Fig. 1. Structure of subroutines solving linear equations.

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
 DATAACS, DATBCS, SORDCS, REDUCS, SOLVCS
 DATAACN, DATBCN, SORDCN, REDUCN, SOLVCN
 SYMCOLR, SYMCOLC

Input/Output: None

Restrictions: $N \geq 1$, $LES \geq LF$

Flags: IAR

C	PROGRAM EXAMP1(OUTPUT,TAPE6=OUTPUT)	000001
C	INTEGER ITAG(30),LCOL(11),LNXT(30),NOZE(10),NSEQ(10)	000002
	REAL CE(30),V(10),RE(30),DE(10)	000003
C	THIS IS THE MAIN PROGRAM FOR SOLVING THE SYSTEM OF LINEAR	000004
C	EQUATIONS DESCRIBED IN EXAMPLE 1 USING THE PACKAGE SSLE	000005
C	DATA LCOL/1,2,4,5,7,11,12,13,14,16,17/	000006
C	DATA ITAG/2,1,3,4,7,9,2,3,4,6,9,8,7,6,10,8,14*0/	000007
C	DATA CE/-.8,1.,.1,1.,-.3,.3,1.,-.5,.1,.5,0.,.1,.1,.3,-1.,.2,14*0/	000008
C	DATA DE/2.,2.,10.,2.,7.2,3.,2.,10.,2.,3./	000009
C	DATA V/0.,1.,0.,0.,1.,0.,-2.,0.,0.,3./	000010
	DATA N/10/,LES/30/	000011
	IAR=2	000012
	WRITE(6,60)	000013
	CALL SECOND(TM1)	000014
	CALL SSLERN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR)	000015
	IF(IAR.EQ.1)STOP	000016
	CALL SECOND(TM2)	000017
	CPU=TM2-TM1	000018
	WRITE(6,70)CPU	000019
70	FORMAT(/," CPU TIME:",F8.3," SECONDS")	000020
60	FORMAT(1H,"EXAMPLE 1",/)	000021
	WRITE(6,100)	000022
	IF(N.NE.4.AND.N.NE.2)WRITE(6,120)	000023
	IF(N.EQ.4.OR.N.EQ.2)WRITE(6,130)	000024
	N3=(N+2)/3	000025
	DO 2 I=1,N3	000026
	LI=I	000027
	LO=I+2*N3	000028
	IF(LO.GT.N)LO=I+N3	000029
2	WRITE(6,110)(J,V(J),J=LI,LO,N3)	000030
130	FORMAT(1H," VARIABLE VALUE VARIABLE VALUE",/)	000031
120	FORMAT(1H," VARIABLE VALUE VARIABLE VALUE VARIABLE	000032
	+E VALUE",/)	000033
110	FORMAT(1X,4(16,2X,E15.8))	000034
100	FORMAT(1H ,/, " SOLUTION",/)	000035
	STOP	000036
	END	000037
		000038
		000039
		000040
		000041
		000042

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.

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
DATA CN	31	76	39
DATBCN	45	154	40
SYMCOLC	94	217	41
Total	1527	5756	


```
L=LCOL(K) 000196
I=ITAG(L) 000197
60 IF (I-IP) 70,80,90 000198
70 L=LNXT(L) 000199
IF (L.LE.0) GO TO 100 000200
I=ITAG(L) 000201
GO TO 60 000202
C 000203
C ELEMENTS OF REDUCED MATRIX A(J) 000204
C 000205
80 CE(L)=CE(L)-CF*CE(LP) 000206
L=LNXT(L) 000207
IF (L.LE.0) GO TO 100 000208
I=ITAG(L) 000209
90 LI=LNXT(LI) 000210
IF (LI.LE.0) GO TO 100 000211
IP=ITAG(LI) 000212
LP=LNXT(LP) 000213
GO TO 60 000214
100 LK=LNXT(LK) 000215
IF (LK.GT.0) GO TO 50 000216
110 CONTINUE 000217
RETURN 000218
120 WRITE (6,130) 000219
IAR=1 000220
130 FORMAT (" DIMENSIONED AREA TOO SMALL IN REDURS") 000221
RETURN 000222
END 000223
C 000224
C SUBROUTINE SOLVRS (N,LCOL,NSEQ,ITAG,LNXT,CE,V) 000225
C 000226
C THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF 000227
C 000228
C MATRIX MULTIPLICATIONS 000229
C 000230
C INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1) 000231
C REAL CE(1),V(1),CF,SUM 000232
C 000233
C FIRST PART OF THE SOLUTION 000234
C B1=L(N)*L(N-1)*...*L(1)*B. 000235
C 000236
C 000237
DO 20 J=1,N 000238
K=NSEQ(J) 000239
CF=V(K) 000240
V(K)=0. 000241
L=LCOL(K) 000242
10 I=ITAG(L) 000243
V(I)=V(I)+CE(L)*CF 000244
L=LNXT(L) 000245
IF (L.GT.0) GO TO 10 000246
20 CONTINUE 000247
C 000248
C SECOND PART OF THE SOLUTION 000249
C X=R(1)*R(2)*...*R(N)*B1 000250
C 000251
N1=N-1 000252
DO 40 JJ=1,N1 000253
J=N-JJ 000254
K=NSEQ(J) 000255
SUM=V(K) 000256
L=LCOL(K) 000257
30 I=ITAG(L) 000258
IF (I.NE.K) SUM=SUM+CE(L)*V(I) 000259
L=LNXT(L) 000260
```

```
IF (L.CT.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
C INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
C REAL CE(1)
C LF=LCOL(N+1)
C DO 10 I=1,LES
C LNXT(I)=I+1
10 CONTINUE
C LNXT(LES)=0
C DO 20 I=LF,LES
C CE(I)=0.
20 CONTINUE
C
C CALCULATE MATRICES NSEQ 8CLNXT
C
C DO 50 I=1,N
C J=I+1
C IF (LCOL(I).EQ.0) GO TO 50
30 IF (LCOL(J).NE.0) GO TO 40
C J=J+1
C 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
C INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
C REAL CE(1)
C
C IF IAR.EQ.2 CHECK SYMMETRY
C
C IF (IAR.EQ.2) CALL SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C CALCULATE MATRIX NOZE
C
C DO 20 I=1,N
C NOZE(I)=0
C NE=LCOL(I)
10 IF (NE.EQ.0) GO TO 20
C NOZE(I)=NOZE(I)+1
C NE=LNXT(NE)
C GO TO 10
20 CONTINUE
RETURN
END
C
C
```



```
C          CALL DATBRN (N, LCOL, NOZE, NSEQ, ITAG, LNXT, CE, RE, LF, LES, IAR)          000391
C          IAR=0          000392
C          CALL SORDRN (N, LCOL, NOZE, NSEQ, ITAG, LNXT, CE, RE, LF, IAR)          000393
C          IF (IAR.EQ.1) RETURN          000394
C          CALL REDURN (N, LCOL, NSEQ, ITAG, LNXT, CE, RE, DE, LF, IAR)          000395
C          IF (IAR.EQ.1) RETURN          000396
C          CALL SOLVRN (N, LCOL, NSEQ, ITAG, LNXT, CE, RE, DE, V)          000397
C          RETURN          000398
C          END          000399
C          000400
C          000401
C          000402
C          SUBROUTINE SORDRN (N, LCOL, NOZE, NSEQ, ITAG, LNXT, CE, RE, LF, IAR)          000403
C          THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION          000404
C          FOR NONSYMMETRICAL MATRICES          000405
C          000406
C          000407
C          000408
C          INTEGER LCOL(1), NOZE(1), NSEQ(1), ITAG(1), LNXT(1)          000409
C          REAL CE(1), RE(1)          000410
C          N1=N-1          000411
C          DO 140 J=1, N1          000412
C          K=NSEQ(J)          000413
C          MIN=NOZE(K)          000414
C          M=J          000415
C          J1=J+1          000416
C          000417
C          FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS          000418
C          000419
C          DO 10 I=J1, N          000420
C          K=NSEQ(I)          000421
C          IF (NOZE(K).GE.MIN) GO TO 10          000422
C          MIN=NOZE(K)          000423
C          M=I          000424
C          10 CONTINUE          000425
C          000426
C          INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M          000427
C          000428
C          KP=NSEQ(M)          000429
C          NSEQ(M)=NSEQ(J)          000430
C          NSEQ(J)=KP          000431
C          LK=LCOL(KP)          000432
C          20 IF (LK.LE.0) GO TO 140          000433
C          000434
C          TAKE THE NEXT ELEMENT FROM MINIMAL COLUMN M          000435
C          IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE          000436
C          000437
C          K=ITAG(LK)          000438
C          LA=0          000439
C          LI=LCOL(KP)          000440
C          IP=ITAG(LI)          000441
C          L=LCOL(K)          000442
C          I=ITAG(L)          000443
C          30 IF (I-IP) 50, 40, 90          000444
C          40 LA=L          000445
C          L=LNXT(L)          000446
C          IF (L.GT.0) I=ITAG(L)          000447
C          IF (L.LE.0) I=N+1          000448
C          GO TO 100          000449
C          50 IF (I.NE.KP) GO TO 60          000450
C          LN=LNXT(L)          000451
C          IF (LA.GT.0) LNXT(LA)=LN          000452
C          IF (LA.LE.0) LCOL(K)=LN          000453
C          LNXT(L)=LF          000454
C          LF=L          000455
```

	CE(L)=0.	000456
	RE(L)=0.	000457
	NOZE(K)=NOZE(K)-1	000458
	L=LN	000459
	GO TO 70	000460
60	LA=L	000461
	L=LNXT(L)	000462
70	IF (L.LE.0) GO TO 80	000463
	I=ITAG(L)	000464
	GO TO 30	000465
80	IF (LI.LE.0) GO TO 120	000466
	I=N+1	000467
	GO TO 30	000468
90	IF (IP.EQ.K) GO TO 100	000469
	IF (LF.LE.0) GO TO 130	000470
	LN=LF	000471
	IF (LA.GT.0) LNXT(LA)=LN	000472
	IF (LA.LE.0) LCOL(K)=LN	000473
	LF=LNXT(LN)	000474
	LNXT(LN)=L	000475
	ITAG(LN)=IP	000476
	NOZE(K)=NOZE(K)+1	000477
	LA=LN	000478
100	LI=LNXT(LI)	000479
	IF (LI.LE.0) GO TO 110	000480
	IP=ITAG(LI)	000481
	GO TO 30	000482
110	IF (L.GT.0) IP=N+1	000483
	IF (L.GT.0) GO TO 30	000484
120	LK=LNXT(LK)	000485
	GO TO 20	000486
130	WRITE (6,150)	000487
	IAR=1	000488
	RETURN	000489
140	CONTINUE	000490
	RETURN	000491
C	150	000492
	FORMAT (" DIMENSIONED AREA TOO SMALL IN SORDRN")	000493
	END	000494
C		000495
C		000496
	SUBROUTINE REDURN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,LF,IAR)	000497
C		000498
C	THIS SUBROUTINE PERFORMS REDUCTION OF A NONSYMMETRICAL MATRIX	000499
C		000500
C	THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES	000501
C		000502
	INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)	000503
	REAL CE(1),RE(1),DE(1),D,CF,RF	000504
	DO 100 J=1,N	000505
	KP=NSEQ(J)	000506
	D=1/DE(KP)	000507
	DE(KP)=D	000508
	LK=LCOL(KP)	000509
	IF (LK.LE.0) GO TO 100	000510
10	RE(LK)=D*RE(LK)	000511
	LK=LNXT(LK)	000512
	IF (LK.GT.0) GO TO 10	000513
	LK=LCOL(KP)	000514
20	K=ITAG(LK)	000515
		000516
C		000517
C	ELEMENTS OF FACTOR MATRICES L(J) AND R(J)	000518
		000519
	CF=RE(LK)	000520
	RF=CE(LK)	000520

```

      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
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
      B1=L(N)*L(N-1)*...*L(1)*B
C
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
      X=R(1)*R(2)*...*R(N)*B1
C
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) 000586
L=LNXT(L) 000587
GO TO 30 000588
40 V(K)=SUM 000589
50 CONTINUE 000590
RETURN 000591
END 000592

C
C
SUBROUTINE DATARN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR) 000595
C
C THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES: 000596
C
C NSEQ, LNXT, LF 000597
C
C INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1) 000598
C REAL CE(1),RE(1) 000599
C LF=LCOL(N+1) 000600
C DO 10 I=1,LES 000601
C LNXT(I)=I+1 000602
10 CONTINUE 000603
LNXT(LES)=0 000604
DO 20 I=LF,LES 000605
CE(I)=0. 000606
20 CONTINUE 000607

C
C CALCULATE MATRICES NSEQ & LNXT 000608
C
C DO 50 I=1,N 000609
C J=I+1 000610
C IF (LCOL(I).EQ.0) GO TO 50 000611
30 IF (LCOL(J).NE.0) GO TO 40 000612
C J=J+1 000613
C GO TO 30 000614
40 LNXT(LCOL(I+1)-1)=0 000615
50 NSEQ(I)=I 000616
RETURN 000617
END 000618

C
C
SUBROUTINE DATBRN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR) 000626
C
C THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES: 000627
C
C NOZE, RE 000628
C
C INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1) 000629
C REAL CE(1),RE(1) 000630
C
C IF IAR.EQ.2 CHECK SYMMETRY 000631
C
C IF (IAR.EQ.2) CALL SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF) 000632
C
C CALCULATE MATRIX NOZE 000633
C
C DO 20 I=1,N 000634
C NOZE(I)=0 000635
C NE=LCOL(I) 000636
10 IF (NE.EQ.0) GO TO 20 000637
C NOZE(I)=NOZE(I)+1 000638
C NE=LNXT(NE) 000639
C GO TO 10 000640
20 CONTINUE 000641
DO 30 I=1,LES 000642
RE(I)=0. 000643
30 CONTINUE 000644
END 000645
```

```
C      30 CONTINUE                                000651
C      CALCULATE ROWWISE STORED COEFFICIENT MATRIX 000652
C      DO 70 I=1,N                                000653
C      NE=LCOL(I)                                  000654
40     IF (NE.EQ.0) GO TO 70                        000655
C      NR=ITAG(NE)                                  000656
C      LI=LCOL(NR)                                  000657
50     IF (ITAG(LI).EQ.I) GO TO 60                  000658
C      LI=LNXT(LI)                                  000659
C      GO TO 50                                     000660
60     RE(LI)=CE(NE)                                000661
C      NE=LNXT(NE)                                  000662
C      GO TO 40                                     000663
70     CONTINUE                                    000664
C      RETURN                                       000665
C      END                                          000666
C      SUBROUTINE SYMCOLR (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF) 000667
C      THIS SUBROUTINE CHECKS THE STRUCTURAL SYMMETRY AND ORDERING OF 000668
C      COLUMNS OF THE COEFFICIENT MATRIX. IF THE MATRIX IS NOT 000669
C      SYMMETRICAL, NEW ZERO COEFFICIENTS ARE ADDED TO REACH SYMMETRY 000670
C      COLUMNS ARE REORDERED IF NECESSARY 000671
C      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1) 000672
C      REAL CE(1) 000673
C      IS=0 000674
C      DO 70 I=1,N 000675
C      LC=LCOL(I) 000676
C      IF (LC.EQ.0) GO TO 70 000677
10     IR=ITAG(LC) 000678
C      CHECK IF THERE EXISTS AN ELEMENT SYMMETRICAL TO THE ONE 000679
C      CONSIDERED 000680
C      NLC=LCOL(IR) 000681
20     IF (NLC.EQ.0) GO TO 30 000682
C      LLC=NLC 000683
C      LLR=ITAG(LLC) 000684
C      IF (LLR.EQ.I) GO TO 60 000685
C      NLC=LNXT(LLC) 000686
C      GO TO 20 000687
C      MATRIX IS NONSYMMETRICAL-ADD NECESSARY ZERO COEFFICIENTS 000688
C      30 IF (LCOL(IR).NE.0) GO TO 40 000689
C      LCOL(IR)=LF 000690
C      GO TO 50 000691
40     LNXT(LLC)=LF 000692
50     CE(LF)=0. 000693
C      IS=1 000694
C      ITAG(LF)=I 000695
C      NOZE(IR)=NOZE(IR)+1 000696
C      IF=LF 000697
C      LF=LNXT(IF) 000698
C      LNXT(IF)=0 000699
C      PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER 000700
C      CONSIDERATION 000701
C      30 IF (LCOL(IR).NE.0) GO TO 40 000702
C      LCOL(IR)=LF 000703
C      GO TO 50 000704
40     LNXT(LLC)=LF 000705
50     CE(LF)=0. 000706
C      IS=1 000707
C      ITAG(LF)=I 000708
C      NOZE(IR)=NOZE(IR)+1 000709
C      IF=LF 000710
C      LF=LNXT(IF) 000711
C      LNXT(IF)=0 000712
C      PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER 000713
C      CONSIDERATION 000714
C      30 IF (LCOL(IR).NE.0) GO TO 40 000715
```

```

C
60 LC=LNXT(LC) 000716
   IF (LC.NE.0) GO TO 10 000717
70 CONTINUE 000718
   IF (IS.EQ.1) WRITE (6,130) 000719
C 000720
C     ROWS AND COLUMNS ARE ORDERED SUCH THAT INCREASING VALUES OF
C     INDICES DESCRIBE THE NEXT NONZERO ELEMENTS IN EACH ROW AND
C     COLUMN 000721
C 000722
C     IS=0 000723
C     DO 120 I=1,N 000724
C     I1=LCOL(I) 000725
80 I1=I1 000726
90 I2=LNXT(I1) 000727
   IF (I2.EQ.0) GO TO 110 000728
   IF (ITAG(I1).LT.ITAG(I2)) I1=I2 000729
   IF (ITAG(I1).LT.ITAG(I2)) GO TO 90 000730
   IS=1 000731
   IF (I1.EQ.LCOL(I)) GO TO 100 000732
C 000733
C     EXCHANGE TWO ELEMENTS 000734
C 000735
C     LNXT(I0)=I2 000736
C     LNXT(I1)=LNXT(I2) 000737
C     LNXT(I2)=I1 000738
C     I1=I2 000739
C     GO TO 90 000740
100 LCOL(I)=I2 000741
    LNXT(I1)=LNXT(I2) 000742
    LNXT(I2)=I1 000743
    I1=I2 000744
    GO TO 90 000745
C 000746
C     PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C     CONSIDERATION 000747
C 000748
C 110 I0=I1 000749
    I1=LNXT(I1) 000750
    IF (I1.EQ.0) GO TO 120 000751
    GO TO 80 000752
120 CONTINUE 000753
    IF (IS.EQ.1) WRITE (6,140) 000754
    RETURN 000755
C 000756
C 130 FORMAT (1H , " INITIAL COEFFICIENT MATRIX NONSYMMETRICAL") 000757
C 140 FORMAT (1H , " REORDERING OF COLUMNS") 000758
C     END 000759
C 000760
C     SUBROUTINE SSLECS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,V,LF,LES,IAR) 000761
C 000762
C     THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS 000763
C     WHOSE COEFFICIENT MATRIX IS COMPLEX AND SYMMETRICAL 000764
C     BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN 000765
C     " LARGE SPARSE SETS OF LINEAR EQUATIONS " 000766
C     - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF 000767
C     MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970- 000768
C     J.K.REID, EDITOR, ACADEMIC PRESS, 1971. 000769
C 000770
C 000771
C 000772
C 000773
C 000774
C 000775
C 000776
C 000777
C 000778
C 000779
C 000780

```

```
C
INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
COMPLEX CE(1),V(1)
000781
C
C      N      IS THE NUMBER OF UNKNOWN,ORDER OF THE MATRIX
000782
C
C      LCOL STARTING POSITION OF COLUMNS
000783
C      N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
000784
C      POSITION. DIMENSION IS EQUAL TO N+1
000785
C
C      NOZE NUMBER OF NON-ZERO TERMS
000786
C      CALCULATED BY THE DATBCS SUBROUTINE
000787
C      DIMENSION IS EQUAL TO N
000788
C
C      NSEQ SEQUENCE OF PIVOTAL INDICES
000789
C      CALCULATED BY THE DATACS SUBROUTINE
000790
C      DIMENSION IS EQUAL TO N
000791
C
C      ITAG ROW INDEX OF ELEMENTS STORED IN CE
000792
C      DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
000793
C      ELEMENTS
000794
C
C      LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
000795
C      DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
000796
C      ELEMENTS
000797
C      CALCULATED BY THE DATACS SUBROUTINE
000798
C
C      CE      IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
000799
C
C      V      AN ARRAY OF DIMENSION N STORING THE RIGHT HAND
000800
C      SIDE VECTOR
000801
C      AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
000802
C
C      LF      INDICATOR FOR THE NEXT VACANT LOCATION
000803
C      CALCULATED BY THE DATACS SUBROUTINE
000804
C
C      LES     NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
000805
C
C      IAR     INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
000806
C      ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
000807
C      AREA
000808
C
C      CALL DATACS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
000809
C
C      CALL DATBCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR)
000810
C      IAR=0
000811
C      CALL SORDCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
000812
C      IF (IAR.EQ.1) RETURN
000813
C      CALL REDUCS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR)
000814
C      IF (IAR.EQ.1) RETURN
000815
C      CALL SOLVCS (N,LCOL,NSEQ,ITAG,LNXT,CE,V)
000816
C      RETURN
000817
C      END
000818
C
C      SUBROUTINE SORDCS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,IAR)
000819
C
C      THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
000820
C      FOR SYMMETRICAL MATRICES
000821
C
C      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
000822
C      COMPLEX CE(1)
000823
C      N1=N-1
000824
C      DO 140 J=1,N1
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
```

	K=NSEQ(J)	000846
	MIN=NOZE(K)	000847
	M=J	000848
	J1=J+1	000849
C		000850
C	FIND A COLUMN M WITH MINIMUM NUMBER OF ELEMENTS	000851
C		000852
	DO 10 I=J1,N	000853
	K=NSEQ(I)	000854
	IF (NOZE(K).GE.MIN) GO TO 10	000855
	MIN=NOZE(K)	000856
	M=I	000857
	10 CONTINUE	000858
C		000859
C	INTERCHANGE INDICES WITHIN NSEQ FOR ACTUAL J AND M	000860
C		000861
	KP=NSEQ(M)	000862
	NSEQ(M)=NSEQ(J)	000863
	NSEQ(J)=KP	000864
	LK=LCOL(KP)	000865
C		000866
C	TAKE THE SUCCESSFUL ELEMENT FROM MINIMAL COLUMN M	000867
C	IF THIS ELEMENT IS DIAGONAL TAKE THE NEXT ONE	000868
C		000869
	20 K=ITAG(LK)	000870
	IF (K.EQ.KP) GO TO 100	000871
	LA=0	000872
	LI=LCOL(KP)	000873
	IP=ITAG(LI)	000874
	L=LCOL(K)	000875
	I=ITAG(L)	000876
	30 IF (I-IP) 40,50,80	000877
	40 LA=L	000878
	L=LNXT(L)	000879
	IF (L.GT.0) I=ITAG(L)	000880
	IF (L.LE.0) I=N+1	000881
	GO TO 30	000882
	50 IF (I.NE.KP) GO TO 60	000883
	LN=LNXT(L)	000884
	IF (LA.GT.0) LNXT(LA)=LN	000885
	IF (LA.LE.0) LCOL(K)=LN	000886
	LNXT(L)=LF	000887
	LF=L	000888
	CE(L)=CPLX(0.,0.)	000889
	NOZE(K)=NOZE(K)-1	000890
	L=LN	000891
	GO TO 70	000892
	60 LA=L	000893
	L=LNXT(L)	000894
	70 IF (L.GT.0) I=ITAG(L)	000895
	IF (L.LE.0) I=N+1	000896
	GO TO 90	000897
	80 IF (LF.LE.0) GO TO 110	000898
	LN=LF	000899
	IF (LA.GT.0) LNXT(LA)=LN	000900
	IF (LA.LE.0) LCOL(K)=LN	000901
	LF=LNXT(LN)	000902
	LNXT(LN)=L	000903
	ITAG(LN)=IP	000904
	NOZE(K)=NOZE(K)+1	000905
	LA=LN	000906
	90 LI=LNXT(LI)	000907
	IF (LI.LE.0) GO TO 100	000908
	IP=ITAG(LI)	000909
	GO TO 30	000910

```
100 LK=LNXT(LK) 000911
    GO TO 130 000912
110 WRITE (6,120) 000913
    IAR=1 000914
120 FORMAT (" DIMENSIONED AREA TOO SMALL IN SORDCS") 000915
    RETURN 000916
130 IF (LK.GT.0) GO TO 20 000917
140 CONTINUE 000918
    RETURN 000919
    END 000920
C 000921
C 000922
SUBROUTINE REDUCS (N,LCOL,NSEQ,ITAG,LNXT,CE,LF,IAR) 000923
C 000924
C THIS SUBROUTINE PERFORMS REDUCTION OF A SYMMETRICAL MATRIX 000925
C 000926
C THE FINAL MATRIX CE CONTAINS THE SEQUENCE OF R,L MATRICES 000927
C 000928
INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1) 000929
COMPLEX CE(1),D,CF 000930
DO 110 J=1,N 000931
KP=NSEQ(J) 000932
LK=LCOL(KP) 000933
LP=LF 000934
10 IF (LP.LE.0) GO TO 120 000935
K=ITAG(LK) 000936
IF (K.NE.KP) GO TO 20 000937
C 000938
C DIVISION BY DIAGONAL ELEMENT 000939
C 000940
D=1./CE(LK) 000941
CE(LK)=D 000942
GO TO 30 000943
20 CE(LP)=CE(LK) 000944
30 LK=LNXT(LK) 000945
IF (LK.LE.0) GO TO 40 000946
LP=LNXT(LP) 000947
GO TO 10 000948
40 LK=LCOL(KP) 000949
50 K=ITAG(LK) 000950
IF (K.EQ.KP) GO TO 100 000951
C 000952
C ELEMENTS OF FACTOR MATRICES L(J) AND R(J) 000953
C 000954
CF=D*CE(LK) 000955
CE(LK)=-CF 000956
LP=LF 000957
LI=LCOL(KP) 000958
IP=ITAG(LI) 000959
L=LCOL(K) 000960
I=ITAG(L) 000961
60 IF (I-IP) 70,80,90 000962
70 L=LNXT(L) 000963
IF (L.LE.0) GO TO 100 000964
I=ITAG(L) 000965
GO TO 60 000966
C 000967
C ELEMENTS OF REDUCED MATRIX A(J) 000968
C 000969
80 CE(L)=CE(L)-CF*CE(LP) 000970
L=LNXT(L) 000971
IF (L.LE.0) GO TO 100 000972
I=ITAG(L) 000973
90 LI=LNXT(LI) 000974
IF (LI.LE.0) GO TO 100 000975
```

```
IP=ITAG(LI) 000976
LP=LNXT(LP) 000977
GO TO 60 000978
100 LK=LNXT(LK) 000979
IF (LK.GT.0) GO TO 50 000980
110 CONTINUE 000981
RETURN 000982
120 WRITE (6,130) 000983
IAR=1 000984
130 FORMAT (" DIMENSIONED AREA TOO SMALL IN REDUCS") 000985
RETURN 000986
END 000987
C 000988
C 000989
SUBROUTINE SOLVCS (N,LCOL,NSEQ,ITAG,LNXT,CE,V) 000990
C 000991
C THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF 000992
C 000993
C MATRIX MULTIPLICATIONS 000994
C 000995
C INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1) 000996
C COMPLEX CE(1),V(1),CF,SUM 000997
C 000998
C FIRST PART OF THE SOLUTION 000999
C B1=L(N)*L(N-1)*...*L(1)*B. 001000
C 001001
DO 20 J=1,N 001002
K=NSEQ(J) 001003
CF=V(K) 001004
V(K)=CMLPX(0.,0.) 001005
L=LCOL(K) 001006
10 I=ITAG(L) 001007
V(I)=V(I)+CE(L)*CF 001008
L=LNXT(L) 001009
IF (L.GT.0) GO TO 10 001010
20 CONTINUE 001011
C 001012
C SECOND PART OF THE SOLUTION 001013
C X=R(1)*R(2)*...*R(N)*B1 001014
C 001015
N1=N-1 001016
DO 40 JJ=1,N1 001017
J=N-JJ 001018
K=NSEQ(J) 001019
SUM=V(K) 001020
L=LCOL(K) 001021
30 I=ITAG(L) 001022
IF (I.NE.K) SUM=SUM+CE(L)*V(I) 001023
L=LNXT(L) 001024
IF (L.GT.0) GO TO 30 001025
V(K)=SUM 001026
40 CONTINUE 001027
RETURN 001028
END 001029
C 001030
C 001031
SUBROUTINE DATACS (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF,LES,IAR) 001032
C 001033
C THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES: 001034
C 001035
C NSEQ, LNXT, LF 001036
C 001037
C INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1) 001038
C COMPLEX CE(1) 001039
C LF=LCOL(N+1) 001040
```

```

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
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
C      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
C      COMPLEX CE(1)
C
C      IF IAR.EQ.2 CHECK SYMMETRY
C
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
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      WHOSE COEFFICIENT MATRIX IS SPARSE AND NONSYMMETRICAL
C      BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C      " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C      - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C      MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C      J.K.REID, EDITOR, ACADEMIC PRESS, 1971.

```

```

001041
001042
001043
001044
001045
001046
001047
001048
001049
001050
001051
001052
001053
001054
001055
001056
001057
001058
001059
001060
001061
001062
001063
001064
001065
001066
001067
001068
001069
001070
001071
001072
001073
001074
001075
001076
001077
001078
001079
001080
001081
001082
001083
001084
001085
001086
001087
001088
001089
001090
001091
001092
001093
001094
001095
001096
001097
001098
001099
001100
001101
001102
001103
001104
001105

```



```
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 TOO 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
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)
```

001236
001237
001238
001239
001240
001241
001242
001243
001244
001245
001246
001247
001248
001249
001250
001251
001252
001253
001254
001255
001256
001257
001258
001259
001260
001261
001262
001263
001264
001265
001266
001267
001268
001269
001270
001271
001272
001273
001274
001275
001276
001277
001278
001279
001280
001281
001282
001283
001284
001285
001286
001287
001288
001289
001290
001291
001292
001293
001294
001295
001296
001297
001298
001299
001300

	GO TO 80	001301
70	IF (IP.NE.K) GO TO 80	001302
	DE(K)=DE(K)-CF*CE(LI)	001303
80	LI=LNXT(LI)	001304
	IF (LI.LE.0) GO TO 90	001305
	IP=ITAG(LI)	001306
	GO TO 40	001307
90	LK=LNXT(LK)	001308
	IF (LK.GT.0) GO TO 20	001309
100	CONTINUE	001310
	RETURN	001311
	END	001312
C		001313
C		001314
	SUBROUTINE SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)	001315
C		001316
C	THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF	001317
C		001318
C	MATRIX MULTIPLICATIONS	001319
C		001320
	INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)	001321
	COMPLEX CE(1),RE(1),DE(1),V(1),CF,SUM	001322
C		001323
C	FIRST PART OF THE SOLUTION	001324
C	B1=L(N)*L(N-1)*...*L(1)*B	001325
C		001326
	DO 20 J=1,N	001327
	K=NSEQ(J)	001328
	CF=DE(K)*V(K)	001329
	V(K)=CF	001330
	L=LCOL(K)	001331
10	IF (L.LE.0) GO TO 20	001332
	I=ITAG(L)	001333
	V(I)=V(I)-CE(L)*CF	001334
	L=LNXT(L)	001335
	GO TO 10	001336
20	CONTINUE	001337
C		001338
C	SECOND PART OF THE SOLUTION	001339
C	X=R(1)*R(2)*...*R(N)*B1	001340
C		001341
	N1=N-1	001342
	DO 50 JJ=1,N1	001343
	J=N-JJ	001344
	K=NSEQ(J)	001345
	SUM=V(K)	001346
	L=LCOL(K)	001347
30	IF (L.LE.0) GO TO 40	001348
	I=ITAG(L)	001349
	SUM=SUM-RE(L)*V(I)	001350
	L=LNXT(L)	001351
	GO TO 30	001352
40	V(K)=SUM	001353
50	CONTINUE	001354
	RETURN	001355
	END	001356
C		001357
C		001358
	SUBROUTINE DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)	001359
C		001360
C	THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:	001361
C		001362
C	NSEQ, LNXT, LF	001363
C		001364
	INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)	001365

	COMPLEX CE(1),RE(1)	001366
	LF=LCOL(N+1)	001367
	DO 10 I=1,LES	001368
	LNXT(I)=I+1	001369
10	CONTINUE	001370
	LNXT(LES)=0	001371
	DO 20 I=LF,LES	001372
	CE(I)=CMLX(0.,0.)	001373
20	CONTINUE	001374
C		001375
C	CALCULATE MATRICES NSEQ & LNXT	001376
C		001377
	DO 50 I=1,N	001378
	J=I+1	001379
	IF (LCOL(I).EQ.0) GO TO 50	001380
30	IF (LCOL(J).NE.0) GO TO 40	001381
	J=J+1	001382
	GO TO 30	001383
40	LNXT(LCOL(I+1)-1)=0	001384
50	NSEQ(I)=I	001385
	RETURN	001386
	END	001387
C		001388
C		001389
	SUBROUTINE DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)	001390
C		001391
C	THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:	001392
C		001393
C	NOZE,RE	001394
C		001395
	INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)	001396
	COMPLEX CE(1),RE(1)	001397
C		001398
C	IF IAR.EQ.2 CHECK SYMMETRY	001399
C		001400
	IF (IAR.EQ.2) CALL SYMCOLC (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)	001401
C		001402
C	CALCULATE MATRIX NOZE	001403
C		001404
	DO 20 I=1,N	001405
	NOZE(I)=0	001406
	NE=LCOL(I)	001407
10	IF (NE.EQ.0) GO TO 20	001408
	NOZE(I)=NOZE(I)+1	001409
	NE=LNXT(NE)	001410
	GO TO 10	001411
20	CONTINUE	001412
	DO 30 I=1,LES	001413
	RE(I)=CMLX(0.,0.)	001414
30	CONTINUE	001415
C		001416
C		001417
C	CALCULATE ROWWISE STORED COEFFICIENT MATRIX	001418
		001419
	DO 70 I=1,N	001420
	NE=LCOL(I)	001421
40	IF (NE.EQ.0) GO TO 70	001422
	NR=ITAG(NE)	001423
	LI=LCOL(NR)	001424
50	IF (ITAG(LI).EQ.I) GO TO 60	001425
	LI=LNXT(LI)	001426
	GO TO 50	001427
60	RE(LI)=CE(NE)	001428
	NE=LNXT(NE)	001429
	GO TO 40	001430
70	CONTINUE	001430

```
RETURN 001431
END 001432
C 001433
C 001434
SUBROUTINE SYMCOLC (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF) 001435
C 001436
C THIS SUBROUTINE CHECKS THE STRUCTURAL SYMMETRY AND ORDERING OF 001437
C 001438
C COLUMNS OF THE COEFFICIENT MATRIX. IF THE MATRIX IS NOT 001439
C 001440
C SYMMETRICAL, NEW ZERO COEFFICIENTS ARE ADDED TO REACH SYMMETRY 001441
C 001442
C COLUMNS ARE REORDERED IF NECESSARY 001443
C 001444
INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1) 001445
COMPLEX CE(1) 001446
IS=0 001447
DO 70 I=1,N 001448
LC=LCOL(I) 001449
IF (LC.EQ.0) GO TO 70 001450
10 IR=ITAG(LC) 001451
C 001452
C CHECK IF THERE EXISTS AN ELEMENT SYMMETRICAL TO THE ONE 001453
C 001454
C CONSIDERED 001455
C 001456
NLC=LCOL(IR) 001457
20 IF (NLC.EQ.0) GO TO 30 001458
LLC=NLC 001459
LLR=ITAG(LLC) 001460
IF (LLR.EQ.I) GO TO 60 001461
NLC=LNXT(LLC) 001462
GO TO 20 001463
C 001464
C MATRIX IS NONSYMMETRICAL-ADD NECESSARY ZERO COEFFICIENTS 001465
C 001466
30 IF (LCOL(IR).NE.0) GO TO 40 001467
LCOL(IR)=LF 001468
GO TO 50 001469
40 LNXT(LLC)=LF 001470
50 CE(LF)=CMPLX(0.,0.) 001471
IS=1 001472
ITAG(LF)=I 001473
NOZE(IR)=NOZE(IR)+1 001474
IF=LF 001475
LF=LNXT(IF) 001476
LNXT(IF)=0 001477
C 001478
C PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER 001479
C 001480
C CONSIDERATION 001481
C 001482
60 LC=LNXT(LC) 001483
IF (LC.NE.0) GO TO 10 001484
70 CONTINUE 001485
IF (IS.EQ.1) WRITE (6,130) 001486
C 001487
C ROWS AND COLUMNS ARE ORDERED SUCH THAT INCREASING VALUES OF 001488
C 001489
C INDICES DESCRIBE THE NEXT NONZERO ELEMENTS IN EACH ROW AND 001490
C 001491
C COLUMN 001492
C 001493
IS=0 001494
DO 120 I=1,N 001495
I1=LCOL(I) 001496
80 I1=I1 001497
90 I2=LNXT(I1) 001498
IF (I2.EQ.0) GO TO 110 001499
```

```
IF (ITAG(I1).LT.ITAG(I2)) I1=I2 001496
IF (ITAG(I1).LT.ITAG(I2)) GO TO 90 001497
IS=1 001498
IF (I1.EQ.LCOL(I)) GO TO 100 001499
C 001500
C EXCHANGE TWO ELEMENTS 001501
C 001502
LNXT(I0)=I2 001503
LNXT(I1)=LNXT(I2) 001504
LNXT(I2)=I1 001505
I1=I2 001506
GO TO 90 001507
100 LCOL(I)=I2 001508
LNXT(I1)=LNXT(I2) 001509
LNXT(I2)=I1 001510
I1=I2 001511
GO TO 90 001512
C 001513
C PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER 001514
C CONSIDERATION 001515
C 001516
110 I0=I1 001517
I1=LNXT(I1) 001518
IF (I1.EQ.0) GO TO 120 001519
GO TO 80 001520
120 CONTINUE 001521
IF (IS.EQ.1) WRITE (6,140) 001522
RETURN 001523
C 001524
130 FORMAT (1H ," INITIAL COEFFICIENT MATRIX NONSYMMETRICAL") 001525
140 FORMAT (1H ," REORDERING OF COLUMNS") 001526
001527
```

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.

