

The Naval Ship Research and Development Center is a U. S. Navy center for laboratory effort directed at achieving improved sea and air vehicles. It was formed in March 1967 by merging the David Taylor Model Basin at Carderock, Maryland with the Marine Engineering Laboratory at Annapolis, Maryland.

Naval Ship Research and Development Center Bethesda, Md. 20034

#### MAJOR NSRDC ORGANIZATIONAL COMPONENTS



NDW-NSRDC 3960/44 (REV, 8/71) GPO 917-872

#### UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION	READ INSTRUCTIONS	
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4496		
4. TITLE (and Subtitie)		5. TYPE OF REPORT & PERIOD COVERED
The Performance on the CDC 6400 o	f a Rheinboldt-	Research and Development
Mesztenyi Program for Solving Lar	ge Sparse	
Symmetric Systems of Linear Equat	ions	4496
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Donald A. Gignac		
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Naval Ship Research and Developme	nt Center	SR 014 03 01, Task 15322
Bethesda, Maryland 20034		Work Unit 1-1844-044
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
		July 1974
		13. NUMBER OF PAGES
M HOWTODING ACENCY NAME & ADDDESS/// dillored	from Controlling Office)	25 15 SECURITY CLASS (of this report)
A MONITORING AGENCY NAME & ADDRESS(II differen		
		UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING
		SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release: dist	ribution unlimit	ted
11 1		
17. DISTRIBUTION STATEMENT (of the abstract entered i	n Block 20, il different fro	n Report)
	<u>,</u>	
19. KEY WORDS (Continue on reverse side if necessary and	identify by block number)	
Out-of-Core Solution		
Symmetric System		
Linear Equation Solver		
Arc-graph theory		
20 ABSTRACT (Continue on reverse side if necessary and	identify by block number)	
On the CDC 6400	- EODTRAN E	ad warning of the District 11
Mesztenyi computer program for col	S FURIAR EXLENC	metric matrix equations was
tostod with rospect to contain for so	rving sparse sym	meetic matrix equations was
analysis problems This present	mpre problems re	to be competitive with
CSKVDC another linear equation of	aves not appear	coveral special features in
the CDC 6/00 instruction act recu	uiver. Lack OF Ite in high ower	beed for manipulating the
data structures used	res in night over	head for maniputating the
uala SLIUCLUIES USEU.		

DD 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601 UNCLASSIFIED

JECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

i

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## TABLE OF CONTENTS

	Page
ABSTRACT	1
ADMINISTRATIVE INFORMATION	1
INTRODUCTION	2
THEORY	3
EXAMPLES	4
OBSERVATIONS AND CONCLUSIONS	6
ACKNOWLEDGMENTS	13
APPENDIX - PROGRAM LISTINGS	14

# LIST OF TABLES

Table l.	Solution Times for $A_L^1 X = B$	8
Table 2.	Solution Times for $A_L^2 X = B$	9
Table 3.	CSKYDG Solution Times for $A_L^1 X = B$	10
Table 4.	CSKYDG Solution Times for $A_L^2 X = B$	11
Table 5.	A Final Comparison	12

·

#### ABSTRACT

On the CDC 6400 computer this FORTRAN Extended version of the Rheinboldt-Mesztenyi computer program for solving sparse symmetric matrix equations was tested with respect to certain sample problems representative of structural analysis problems. This program does not appear to be competitive with CSKYDG, another linear equation solver. Lack of several special features in the CDC 6400 instruction set results in high overhead for manipulating the data structures used.

#### ADMINISTRATIVE INFORMATION

The Naval Ship Systems Command (0311) sponsored this study under Subproject SR 014 03 01, Task 15322. The work was performed under work unit number 1-1844-004.

#### INTRODUCTION

In the finite element approach to static structural analysis, the computation of the solution of the equation

KU = P

a positive definite system of simultaneous linear equations, is basic. However, the order of K is often so large that it does not suffice merely to take advantage of K's symmetry and banded structure. For such K it is necessary to consider an out-of-core solution or to devise some storage scheme which exploits sparsity more fully, or even to utilize both of these approaches together. The Rheinboldt-Mesztenyi program<sup>1</sup> was investigated as part of an effort to develop an improved solution capability for large sparse K. This program utilizes a data structure for storing sparse matrices based on arc-graph theory to facilitate an in-core triangular decomposition solution of KU = P for such a K. The best of such programs encountered will eventually be incorporated into large-scale structural programs such as NASTRAN.

The author was recently asked for an opinion on the usefulness of the program of Professors Rheinboldt and Mesztenyi for solving the sparse matrix equations arising in structural analysis calculations on the CDC 6000 series computers. A FORTRAN version of the Rheinboldt-Mesztenyi (R-M) program for symmetric matrices was obtained and modified for the CDC 6400. This report discusses a comparison of this FORTRAN implementation of the R-M program with another FORTRAN linear equation solver, CSKYDG,<sup>2</sup> for certain sample problems.

<sup>&</sup>lt;sup>1</sup> Rheinboldt, W. and Mesztenyi, C., "Problems for the Solution of Large Sparse Matrix Problems Based on the Arc-Graph Structure," University of Maryland Computer Science Center, Technical Report TR-262, September 1973.

<sup>&</sup>lt;sup>2</sup> Gignac, D.A., "CSKYDG, An Out-of-Core Cholesky Algorithm Equation Solver for Large Positive Definite Systems of Linear Equations," Naval Ship Research and Development Center Report 4377, February 1974.

#### THEORY

The R-M program solves the system KU = P using that form of triangular decomposition which does not require square roots. As the decomposition proceeds the rows and corresponding columns of K are interchanged in accordance with the pivoting strategy of Curtis and Reid.<sup>3</sup> If Q represents the permutation matrix for the required row interchanges, then

$$QKQ^T = LDL^T$$

where L is a unit lower triangular matrix and D is a diagonal matrix. We then solve the triangular systems

$$LX_{1} = QP$$
$$DX_{2} = X_{1}$$
$$L^{T}X_{3} = X_{2}$$

using forward or backward substitution as required, and finally obtain

$$U = Q^T X_3$$

In theory this procedure can solve KU = P for any **non**-singular symmetric K. However, the method of Cholesky (square root method) works better for positive definite K. The Cholesky algorithm factors K into the product of a lower triangular matrix S and its transpose, that is,

$$K = SS^T$$
,

then solves the triangular systems

$$SX = P$$
  
 $S^{T}U = X$ 

using forward or backward substitution as required.

The Cholesky algorithm has two advantages over the LDL<sup>T</sup> procedure. First it does not require pivoting to ensure stability, making the matrix

<sup>&</sup>lt;sup>3</sup> Curtis, A.R., and Reid, J.K., "FORTRAN subroutines for the solution of sparse sets of linear equations," United Kingdom Atomic Energy Research Establishment, Harwell, England, Tech. Report AERE-R6844, 1971.

decomposition and the forward and backward substitutions less involved. Secondly the forward substitution can be readily incorporated into the Cholesky decomposition with a significant saving of time. (This last advantage may be realized only for a single solution of KU = P for a given K.) The CSKYDG program takes advantage of these features. Wilkinson and Reinsch<sup>4</sup> give details of both procedures.

#### EXAMPLES

Professor Mesztenyi provided the author with a FORTRAN implementation of the R-M program for the UNIVAC 1108. This program consisted of three subroutines: READ, LU, and SOLVE. The SETUP subroutine was added to facilitate the input of the matrix element. Its two arguments are NS and W. The SETUP subroutine reads from tape 4 the non-zero elements in the lines of the upper triangular half of the coefficient matrix K in the form of triplets

#### I, J, K(K,J)

and writes these triplets in batches of NS on tape 5. If the number of triplets is not a multiple of NS, then the last batch of triplets is filled out to NS elements by adding the appropriate number of triplets

#### 1,0,0.0

The righthand side of KU = P is read from tape 4 and passed through the argument W. The READ subroutine then reads the triplets in batches of NS from tape 5 and sets up arcs of non-zero elements. The LU subroutine then obtains the LDL<sup>T</sup> decomposition of K in terms of these arcs. The SOLVE subroutine then obtains U from P by solving the intermediate systems of equations. The integer packing and unpacking subroutines IPACK and IUNPK were added later.

<sup>&</sup>lt;sup>4</sup> Wilkinson, J.H. and Reinsch, C., editors, "Handbook for Automatic computation," vol. II, "Linear Algebra," Springer-Verlag, New York 1971, pp. 9-11.

After the CDC 6400 version of the R-M program had been checked out, it was compared with CSKYDG,<sup>2</sup> the author's own previously developed linear equation solver for KU = P. The following examples were chosen as the basis for this comparison because these are in some sense representative of systems which arise in structural analysis.

The matrix family of the first example in Table 1,  $A_N^1$ , is generated as follows: Let N be an integer  $\geq 3$ . Let  $C_N$  be the tridiagonal of order N with 4's on the diagonal and a line of -1's above and below the diagonal. Let  $I_N$  be the identity matrix of order N. An (N+1)-banded matrix of order  $N^2$ ,  $A_N^1$  is constructed by

(1) stringing N  $C_N$  submatrices along the diagonal,

(2) inserting lines of N-l  $-I_N$  submatrices above and below the diagonal, and

(3) setting the remaining elements of  $A_N^1$  equal to 0.

The right-hand side of the system  $A_N^1 X = B$  is chosen such that the exact solution X has all components equal to 1.

The matrix family of the second example in Table 2,  $A_N^2$ , is similarly generated. This time let  $C_N$  have diagonal elements of 6. An (N<sup>2</sup>+1)-banded matrix of order N<sup>3</sup> is constructed by

(1) stringing  $N^2$  C<sub>N</sub> submatrices along the diagonal,

(2) inserting lines of N<sup>2</sup>-1 -I  $_{\rm N}$  submatrices above and below the diagonal,

(3) inserting lines of N<sup>2</sup>-N -I $_{N}$  submatrices as the N<sup>th</sup> lines above and below the diagonal, and

(4) setting the remaining elements of  $A_N^2$  equal to 0.

The right-hand side of the system  $A_N^2 X = B$  is chosen such that the exact solution is  $\ell_1$ . These two matrix families have characteristics of matrices arising from finite difference approximations to the Dirichlet problem.

#### **OBSERVATIONS AND CONCLUSIONS**

The tables assembled in this section present the data concerning the performance of the present FORTRAN version of the R-M program on the CDC 6400. In these tables, N indicates the order of the matrix and M its bandwidth. The information in Tables 3 and 4 has been published previously.<sup>2</sup> Note that the SETUP subroutine and the NS parameter of Table 3 are different from those of Tables 1 and 2.

The CDC 6400 version of the R-M program was able to handle the order 225, bandwidth 16 case but not the order 400, bandwidth 21 case of the first matrix family using a field length of 153,400 (Table 1a). Making use of the integer packing and unpacking subroutines IPACK and IUNPK\* to realize a fourfold compression of certain integer arrays, the 'packed' version of the R-M program handled the order 900, bandwidth 31 case but not the order 1225, bandwidth 36 case of the first matrix family using a field length of 145,500 (Table 1b). Similarly the 'unpacked' version of the R-M program handled the order 216, bandwidth 37 case of the second matrix family using a field length of 153,400 (Table 2a). The 'packed' version of the R-M program handled the order 512, bandwidth 65 case of the second matrix family using a field length of 145,500.

Neither the 'packed' or 'unpacked' CDC 6400 version of the R-M program seems competitive in solution times with existing FORTRAN linear equation solvers, in particular CSKYDG, for the examples investigated. Moreover, the CSKYDG program required a field length of only 70,000 to produce the results shown in Tables 3 and 4. However, the accuracy of the solutions of the two programs was comparable.

\*

Those subroutines were provided by Mr. Michael Golden of the Theory of Structures Branch (Code 1844). IPACK packs four integer words (each of which requires no more than fifteen bits) into one word. IUNPK reverses the packing operation. No documentation is available for these subroutines.

The R-M program was at a special disadvantage in this investigation. The "bookkeeping" procedures, so crucial to the R-M program, are most efficient when the program is coded in assembly language (COMPASS for the CDC 6400) rather than FORTRAN. Then the integer packing and unpacking procedures apparently required by the R-M program can be directly implemented in the program rather than using subroutines which are of necessity slower.

Moreover, in discussing the somewhat disappointing performance of the CDC 6400 version of the R-M program, Professor Mesztenyi pointed out that the "fetch" cycle of the CDC 6400 is more expensive than that of the UNIVAC 1108. He also noted that certain FORTRAN compilers appear to provide somewhat inefficient FORTRAN compilations, although the results published here were obtained using that compilation option of the present version of the FTN FORTRAN compiler which usually produces the most efficient code.

It is only fair to note that on the UNIVAC 1108 the performance of the assembly language version of the R-M program is quite satisfactory. For example, the LDL<sup>T</sup> decomposition times listed in the first column of Table 5 for certain members of the first matrix family are UNIVAC 1108 assembly language times taken from Rheinboldt and Mesztenyi (page 48, Table 5). The second column of Table 5 shows the CDC 6400 FORTRAN LDL<sup>T</sup> decomposition times from Table 1b of this report, and the third column of Table 5 gives the corresponding FORTRAN CDC 6400 CSKYDG total solution times from Table 3a of this report. Using the rule of thumb that the UNIVAC 1108 is about three times faster than the CDC 6400, these assembly language results indicate a creditable performance on the part of the R-M program. Even so, CSKYDG (for which, it must be remembered, total solution times are given) still appears to be significantly faster.

# TABLE 1 - R-M SOLUTION TIMES FOR $A_L^1 X = B$

# NS = 250

# A - UNPACKED VERSION

N	м	SETUP (Secs.)	READ (Secs.)	SETUP+READ (Secs.)	LU (Secs.)	SOLVE (Secs.)	LU+SOLVE (Secs.)	TOTAL TIME (Secs.)
25	6	.08	.05	.12	.10	.02	.12	.24
100	11	.14	.14	.28	4.80	.29	5.09	5.37
225	16	.26	.17	.44	16.32	.97	17.29	17.73

ω

## B - PACKED VERSION

N	М	SETUP (Secs.)	READ (Secs.)	SETUP+READ (Secs.)	LU (Secs.)	SOLVE (Secs.)	LU+SOLVE (Secs.)	TOTAL TIME (Secs.)
25	6	.06	.08	.14	.31	.07	.39	.53
100	11	.13	.27	.40	4.43	.64	5.07	5.47
225	16	.25	.54	.80	23.22	2.32	25.53	26.33
400	21	.47	.98	1.44	83.04	6.25	89.29	90.74
625	26	.79	1.53	2.32	198.56	12.39	210.95	213.28
900	31	1.19	2.19	3.39	479.70	25.15	504.8 <b>5</b>	508.24

•

# TABLE 2 - R-M SOLUTION TIMES FOR $A_L^2 X = B$ NS = 250

# A - UNPACKED VERSION

N	М	SETUP (Secs.)	READ (Secs.)	SETUP+READ (Secs.)	LU (Secs.)	SOLVE (Secs.)	LU+SOLVE (Secs.)	TOTAL TIME (Secs.)
125	26	.18	.12	.30	25.23	1.05	26.28	26.58

## **B** - PACKED VERSION

.

N	М	SETUP (Secs.)	READ (Secs.)	SETUP+READ (Secs.)	LU (Secs.)	SOLVE (Secs.)	LU+SOLVE (Secs.)	TOTAL TIME (Secs.)
125	26	.18	.38	.56	27.61	2.48	30.09	30.65
216	37	.36	.70	1.06	170.84	9.41	180.25	181.31
343	50	.62	1.10	1.72	511.02	20.92	531.94	533.66

# TABLE 3 - CSKYDG SOLUTION TIMES FOR $A_L^1 X = B$

NS = 10

N	м	SETUP (Secs.)	SOLUTION (Secs.)	TOTAL TIME (Secs.)
25	6	.07	.15	.21
100	11	.16	.61	.76
225	16	.48	3.07	3.55
400	21	.96	5.69	6.65
625	26	1.92	15.32	17.25
900	31	3.06	22.59	25.64
1225	36	4.87	46.49	51.36
1600	41	6.83	62.39	69.22
2025	46	9.76	108.11	117.87
2500	51	12.69	136.23	148.92
3025	56	17.25	218.10	235.35

.

# TABLE 4 - CSKYDG SOLUTION TIMES FOR $A_L^2 X = B$

NS = 10

N	М	SETUP (Secs.)	SOLUTION (Secs.)	TOTAL TIME (Secs.)
125	26	.39	2.67	3.06
216	37	.80	7.25	8.05
343	50	1.66	17.18	18.84
512	65	3.21	44.19	47.40
729	82	5.67	96.24	101.91
1000	101	9.14	163.59	172.72
1331	122	14.69	347.39	362.08
1728	145	22.54	585.77	608.31

# TABLE 5 - A FINAL COMPARISON

N	м	UNIVAC 1108 ASSEMBLY LDL <sup>T</sup> DECOMPOSITION TIMES (Secs.)	CDC 6400 FORTRAN LDL <sup>T</sup> DECOMPOSITION TIMES (Secs.)	CDC 6400 CSKYDG TOTAL SOLUTION TIMES
100	11	.343	4.43	.76
225	16	1.657	23.22	3.55
400	21	5.921	83.04	6.65

#### ACKNOWLEDGMENTS

The author wishes to thank the following people for their interest and assistance: Professor Charles K. Mesztenyi (University of Maryland Computer Science Center); Dr. Elizabeth H. Cuthill (NSRDC Code 1805); Mr. Michael E. Golden (NSRDC Code 1844); and Dr. Ed Cohen (Naval Ordnance Laboratory).

.

#### APPENDIX - PROGRAM LISTINGS

```
SUBROUTINE SETUP (NS, W)
5
       DIMENSION X(250), II(250), JJ(250), XX(250)
       DIMENSION W(1)
С
      I 01 = 4
      I 02 = 5
      L =0
      READ (IO1) NN, MM
       REWIND IO2
    1 WRITE (IO2)NN
       00 5 I=1,NN
       READ(IOL)(X(K), K=1, MM)
       DO + K=1, MM
       IF (X(K)) 2,4,2
    2 L=L+1
       II(_)=I
       JJ(_) = I + K - 1
      XX(())=X(K)
      IF(L-NS) 4, 3, 4
    3 WRITE (ID2) (II (<K), JJ (KK), XX (KK), KK=1, NS)
      L =0
    4 CONTINUE
    5 CONFINUE
       KLIM = L+1
       DO 5 K=<LIM.NS
       II(\langle )=1
       JJ(K) = J
    6 XX(<)=0.
       WRITE (TJ2)(II(K<), JJ(KK), XX(KK), KK=1, NS)
       REWIND IO2
С
       READ(IO1)(W(K), <=1, NN)
С
       RETURN
```

E ND

	SUBROUTINE READ(NS)	
* * *		*
*		+
#	THE READ SUBROUTINE READS THE NON-ZERO ELEMENTS OF A	*
•	SYNEERIC MAIRIX FRIM UNIT IU, AND SEIS UP THE CORRESPONDING	<b>#</b>
÷.	ARG-GRAPH. THE ELEMENTS ARE READ OF TRIPLETS,	*
-	LIJJOLIJJ HTTA THE 4.0.6. TODIET INDICATING THE END OF INPUT RECORDA	*
Ĩ	ON V THE IPPED TRICINALS SHOLD BE GIVEN TOGETHER WITH THE	+
*	DIAL THE OFFICE STRUCTURES SUBJECT OF COLUMN AND AND AND AND AND AND AND AND AND AN	+
*	DIRJUNA. ELENENIS.	*
* * *		4
С	STORAGE ASSIGNMENT FOR SYMMETRIC DECOMPOSITION	SB00ú200
Ċ	MX - THE MAXIMUM ALLOWED NUMBER OF NON-ZERD COEFFICIENTS,	SB 0 0 0 3 0 0
С	NX - THE MAXIMUM ALLOWED SIZE OF THE MATRIX.	SB000400
	COMMON / DIM/ MX, NX	SB000500
С	ARRAYS OF SIZE M -	SB000700
С	R - INTEGER ARRAY JSED FOR ROW LINKAGE	28000900
С	C - INTEGER ARRAY JSED FOR COLUMN LINKAGE	28000900
С	L - INTEGER ARRAY JSED FOR TAGGING ( D OR 1 )	58001000
C	T - INTEGER ARRAY USED FOR TAGGING ( 0 OR 1 )	58001100
С	B - FL. PT. ARRAY CONTAINING THE VALUES OF THE COEFFICIENTS	20001500
С	THE ARRAYS R,C,L AND T CONTAIN PACKED INLEGERS.	
	CONVON / ARXA YM/ R(5)00), C(5000), L(5000), T(5000), S(2000)	58001500
	INTEGER R, G, T	SB001600
C	ARKAYS UP SIZE N -	58001700
C	IP - INTEGER ARRAY CONTAINS THE SEQUENCE OF FIGURE	SB001800
Ç	ND - INTEGER ARRAY CUNIAINS THE NUMBER OF ELEMENTS	SB001940
C	IN & RUA UP THE UNDECOMPOSED PARTO THE DATATA	SB002000
U C	MAN DE EDUTVALENCED TO TO	SB002100
U.	$\begin{array}{c} \text{HAT} & \text{BE} = (0) \text{VAELANDED} + (0, 1) \text{VAELANDED} + ($	
r	$ \begin{array}{c} COTOR & FART \\ TART & TART \\ TART \\ TART \\ TART \\ TART \\ TART \\ TART$	SB002300
č	M - NIABER OF NONZERO ELEMENTS	SB002400
č	N - STZF OF THE MATRIX	SB002500
č	UP - PIVOT SELECTION PARAMETER	SB 0 0 26 0 0
č	CT1 - MAXIMUM ELEMENT IN THE ORIGINAL MATRIX	SB002700
č	CT2 - MAXIMUM ELEMENT ENCOUNTERED DURING DECOMPOSITION	SB 0 0 2 8 0 0
-	COMMON /DATA/ M, N, CT 1, CT 2	
С		
	DIYENSION II (250), JJ (250), VV (250)	
С		
	M X= 2 0 0 0 J	
	N X=1000	
	I 0= 5	PS001730
	CT1 = 0.0	RS001830
С	LOOP TO ESTABLISH JIAGJNALS	
	KERU (10) N	RS001900
	UU 1U 1=19N D/T) - T	
	r (1) = T	
	T(T) = 0	
	ND(I) = 0	
10	B(I) = ).	RS 0 0 2 5 0 0

•

C LOOP TJ READ ELEMENTS RS002700 LLIM=((N*(N+1))/2)/NS+2 D0 25 LL=1,LLIM REAJ(ID (II(I),JJ(I),VV(I),I=1,NS) C **** MODITY THE FOLLOWING FORMAT AS NEEDED **** RS002900 D0 20 K=1,NS I=II(K) J=JJ(K) V=VV(K) IF (ABS(V).GT.CF1) T1=ABS(V) RS003100 IF (1-J) 45,50,60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT M = M+1 RS003600 RS003600 RS003600 R(I) = M C(J) = M C(M) = C(J) R(I) = M L(M) = 0 T(M) = 0 ND(J) = NJ(J+1) ND(J) = NJ(J+1) ND(J) = NJ(J+1) ND(J) = NJ(I+1) ND(J) = NJ(I+1) ND(J) = NJ(I+1) ND(J) = V ND(I) = NJ(I+1) 20 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 26 EROOF MATIX ELEMENTS 60 RETJRN C ERROR-INSUFFICIENT STORAGE 70 WRITE (6,80) 8J FORMAT (2110 INSJFFICIENT STORAGE) STO <sup>2</sup>		M = N	RS002600
LLIM=((N*(N+1))/2)/NS+2 D0 25 LL=1,LLIM REA3 (10) (TI(1),J)(T),VV(T),I=1,NS) C **** MODIFY THE FOLLOWING FORMAT AS NEEDED **** RS002900 D0 20 K=1,NS I=II(K) J=JJ(K) V=VV(K) IF (ABS(V).GT.CT1) T1=ABS(V) RS003100 C ESTABLISH ARC FOR DFF-DIAGONAL ELEMENT KS003700 R(M) = V RS003800 B(M) = V R(1) = M C(J) = M C(J) = M C(J) = M C(J) = M C(J) = M C(J) = N ND(J) = N) (J+1) S00500 S10RE DIAGONAL S10P C END OF MATRIX ELEMENTS C ERROF-INSUFFICIENT STORAGE S10P S10P S10P C STORE C ERROF-INSUFFICIENT STORAGE) S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P S10P	С	LOOP TO READ ELEMENTS	RS002700
D0 25 LL = 1; LLIM REA3 (10) (II(I), JJ(I), V(I), I= 1; NS) C **** MODITY THE FOLLOWING FORMAT AS NEEDED ***** D0 20 K= 1; NS I=II(K) J=JJ(K) V=VV(K) RS00 3100 IF (I-J) 45, 50, 60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT KS00 3700 RS00 3000 IF (H.GT.MX) GO TO 70 RS00 3000 B(M) = V R(I) = M C(J) = M C(J) = M C(M) = C(J) R(I) = NJ(I+1) ND(J) = NJ(I+1) ND(J) = NJ(I+1) ND(J) = NJ(I+1) 20 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 26 ERROR-INSUFFICIENT STORAGE 70 WRITE (6, 60) S10 St00 S10		LLIH=((N*(N+1))/2)/NS+2	
REA3(ID) (II(I),JJ(I),VV(I),I=1,NS)       RS002900         C       **** MODIFY THE FOLLOWING FORMAT AS NEEDED *****       RS002900         D0 20 K=1,NS       I=II(K)         J=JJ(K)       J=JJ(K)         V=VV(K)       IF(ABS(V).GT.CI1) 3T1=ABS(V)         IF (ABS(V).GT.CI1) 3T1=ABS(V)       RS003100         IF (I-J) 45,50,60       RS003700         C       ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT       RS003800         IF (M.GT.MX) GO TO 70       RS003800         B(M) = V       RS003800         R(M) = R(I)       RS004000         C(M) = C(J)       RS004000         R(M) = R(I)       RS004000         C(M) = C(J)       RS004000         R(M) = N(I+1)       RS004000         NO(I) = ND (I+1)       RS005000         S0 0 10 20       RS004900         C STORE DIAGONAL       RS005100         S0 0 10 20       RS005100         C STORE DIAGONAL       RS005100         S0 0 5100       RS005100         S0 0 CONTINUE       RS005500         C END OF MATRIX ELEMENTS       RS00550		DO 25 LL=1,LLIM	
C **** MODITY THE FOLLOWING FORMAT AS NEEDED **** RS002900 D0 20 K=1,NS I=II (K) J=JJ (K) V=VV(K) IF (ABS(V).GT.CT1) T1=ABS(V) RS003100 IF (I-J) 45,50,50 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT S003300 IF (M.GT.HX) GO TO 70 R(M) = M +1 RS003000 R(M) = V R(M) = R(I) C (Y) = C(J) R(I) = M C (J) = M C (J) = M C (J) = M C (J) = N) (I+1) NO(J) = N) (J+1) NO(J) = N) (J+1) S00 4900 C STORE DIAGONAL S00 500 B (I) = V NO(I) = N) (I+1) 20 COMTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 RETJRN C ERROR-INSU <sup>2</sup> FICIENT STORAGE 70 WRITE (6, 80) S10 <sup>2</sup> S10 <sup>2</sup> RS005500 RS005500		REA)(IO)(II(I),JJ(I),VV(I),I=1,NS)	
D0 20 K=1,NS I=II(K) J=JJ(K) V=VV(K) IF (ABS(V).GT.CT1) T1=ABS(V) RS003100 IF (I-J) 45,50,60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT M = M+1 RS00300 B(M) = V R(M) = R(I) C(M) = C(J) R(I) = M C(J) = M C(J) = M C(J) = M C(J) = N) (J+1) NO(I) = N) (J+1) G0 F0 20 C STORE DIAGONAL STORE DIAGONAL C END OF MATRIX ELEMENTS C END OF MATRIX ELEMENTS C ERROR-INSUFFICIENT STORAGE STOP NDCI (2+0) NDCI = (6,80) STOP STOP RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS00500 RS005	С	**** MODITY THE FOLLOWING FORMAT AS NEEDED ****	RS 0 0 2 9 0 0
I = II ( K) J = JJ ( K) V = VV ( K) IF ( A BS( V).GT.CT 1) T1 = ABS(V) IF ( I - J) 45,50,50 C ESTA BLISH A RC FOR OFF-DIAGONAL ELEMENT R S00 3700 45 M = M+1 R S00 3800 IF ( M.GT. HC) GO TO 70 B (M) = V R (I) = N C (H) = C (J) R (I) = M C (H) = C (J) R (I) = M C (H) = 0 T (H) = 0 NO(I) = NJ (J+1) NO(J) = NJ (J+1) GO TO 20 C STORE DIAGONAL 50 B (I) = V NO(I) = NJ (I+1) 20 CONTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 R ETJRN C ERROR-INSUFFICIENT STORAGE STOP 80 F000 STOP STOP C STOP C ERRORAL (2140 INSJFFIJIENT STORAGE) STOP R S00 5000 R		DO 20 K=1,NS	
J=JJ(K) V=VV(K) IF (ABS(V).GT.CT1) JT1=ABS(V) R (ABS(V).GT.CT1) JT1=ABS(V) R (I -J) 45,50,60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT R S00 3700 45 M = M+1 IF (M.GT.HX) GO TO 70 R (M) = C(I) C (M) = C(I) C (M) = C(I) C (M) = C(I) R (I) = M L (M) = 0 T (M) = 0 ND(I) = NJ (I+1) ND(J) = NJ (J+1) GO TO 20 C STORE JIAGONAL C STORE JIAGONAL C END OF MATRIX ELEMENTS GO RETJRN C END OF MATRIX ELEMENTS GO RETJRN C ERROR-INSUFFICIENT STORAGE STOP STOP STOP STOP STOP STOP STOP STOP STOP C STOP C ERROR-INSUFFICIENT STORAGE STOP STOP STOP R S00 5500 R S00 5600 R R R R R R R R R R R R R R R R R R R		I=II(K)	
V=VV(K) IF (ABS(V).GT.CT1) JT1=ABS(V) IF (I-J) 45,50,60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT M = M+1 RS003800 B(M) = V R(M)=R(I) C(M)=C(J) R(I)=M C(J)=M C(J)=M C(J)=M C(J)=M C(J)=NJ(J+1) ND(J)=NJ(J+1) ND(J)=NJ(J+1) GO FO 20 C STORE JIAGJNAL S0 4000 R (N = V R (N = C) R (N = C) C (J)=N C (J		(X) (L)=(	
IF (ABS(V), GT, GT 1) T1=ABS(V) IF (I-J) 45, 50, 60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT 45 M = M+1 R\$003800 IF (M, GT, MX) GO TO 70 B(M) = V R (M) = R(I) C (M) = C(J) R (I) = M C (J) = M C (J) = M C (J) = M C (J) = N ND(I) = NJ (I+1) ND(J) = NJ (I+1) ND(I) = NJ (I+1) GO TO 20 C STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL C END OF MATRIX ELEMENTS 60 C ERROR-INSUFFICIENT STORAGE STOP ND(I) = NJ (I+J) STORAGE STOP STOP RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000 RS005000		V=VV (K)	
IF (I-J) 45,50,60 C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT RS003700 45 M = M+1 RS003800 IF (M.GT.MX) GO TO 70 R(M) = V R(M) = V R(M) = R(I) C (M) = C(J) R(I) = M C (J) = M C (J) = M C (J) = M C (J) = N ND(I) = N) (I+1) ND(J) = N) (J+1) GO TO 20 C STORE DIAGONAL STOP ND(I) = N) (I+1) 20 CONTINUE 25 CONTINUE 25 CONTINUE C ERROR-INSU <sup>2</sup> FICIENT STORAGE ND(I) = N) (I+1) STORAGE STOP RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800 RS005800		IF (ABS(V).GT.CF1) CT1=ABS(V)	RS003100
C ESTABLISH ARC FOR OFF-DIAGONAL ELEMENT RS003700 45 M = M+1 RS003800 IF (M.GT.MX) GO TO 70 RS003900 B(M) = V RS003900 R(M) = R(I) C(M) = C(J) R(I) = M C(J) = M L(M) = 0 T(M) = 0 ND(I) = NJ (I+1) ND(J) = NJ (J+1) GO TO 20 RS004900 C STORE DIAGONAL RS005000 50 B(I) = V RS005000 50 B(I) = V RS005000 50 B(I) = V RS005000 50 C ONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 RETJRN RS00500 C ERROR-INSUF FIDIENT STORAGE 70 WRITE (6,80) 80 FORMAT (2140 INSJFFIDIENT STORAGE) RS005800 STOP RS005900		IF (I-J) 45,50,60	
45       M = M+1       R\$003800         IF (M+GI-MX) GO TO 70       R\$003900         B(H) = V       R\$004000         R(I) = R(I)       R\$004000         C(H) = C(J)       R(I) = M         L(M) = 0       T(H) = 0         T(H) = 0       R\$004000         R(I) = NJ (I+1)       ND(I) = NJ (J+1)         GO TO 20       R\$004900         C STORE DIAGDNAL       R\$005000         50       B(I) = V         20 CONTINUE       R\$005100         25 CONTINUE       R\$005100         C END OF MATRIX ELEMENTS       R\$005400         GO FO 20       R\$005500         GO FO 20       R\$005500         B(I) = V       R\$005500         S0 B(I) = V       R\$005100         ND(I) = NJ (I+1)       R\$005500         20 CONTINUE       R\$005500         C END OF MATRIX ELEMENTS       R\$005500         GO FORMAT (2140 INSJFFICIENT STORAGE)       R\$005600         70       WRITE (6,80)       R\$005800         80       FORMAT (2140 INSJFFICIENT STORAGE)       R\$005800	С	ESTABLISH ARC FOR OFF-DIAGONAL ELENENT	RS003700
IF (M.GT.MX) GO TO 70 R(M) = V RS003900 B(M) = V R(I) = V RS004000 R(M) = R(I) C(J) = M L(M) = C(J) R(I) = M L(M) = 0 T(Y) = 0 ND(I) = N) (I+1) ND(J) = N) (J+1) GO TO 20 C STORE DIAGONAL SO 50 B(I) = V ND(I) = N) (I+1) 20 CONTINUE 25 CONTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 RETJRN C ERROR-INSUFFICIENT STORAGE RS005400 RS005600	45	M = M+1	RS 0 0 3 8 0 0
B(M) = V R(M) = R(I) C(M) = C(J) R(I) = M C(J) = M L(M) = 0 T(M) = 0 T(M) = 0 ND(I) = NJ(I+1) ND(J) = NJ(J+1) GO TO 20 C STORE DIAGDNAL STORE DIAGDNAL STOP RS00 4900 RS00 5100 RS00 5500 RS00 5500 RS00 5500 RS00 5600 RS00 5800 RS00 5		IF (M.GT.MX) GO TO 70	RS 0 0 3 9 0 0
R (M) = R (I) C (M) = C (J) R (I) = M C (J) = M L (M) = 0 T (M) = 0 T (M) = 0 ND(I) = NJ (J+1) ND(J) = NJ (J+1) G TO 20 C STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL STORE DIAGONAL RS00 4900 RS00 4900 RS00 4900 RS00 4900 RS00 4900 RS00 4900 RS00 4900 RS00 4900 RS00 5000 RS00 5000 RS00 5000 RS00 500 RS00 500 R		B(M) = V	RS004000
C (M) = C (J) R (I) = M C (J) = M L (M) = 0 T (M) = 0 ND(I) = NJ (I+1) ND(J) = NJ (J+1) GO TO 20 C STORE DIAGDNAL STORE DIAGDNAL STORE DIAGDNAL C STORE DIAGDNAL STORE DIAGDNAL STORE DIAGDNAL R S00 5000 R S00 5000 R S00 5000 R S00 5400 R S00 5400 R S00 5500 R S00 5600 STOP R S00 5800 R S00 5800 R S00 5800 R S00 5900		R(M) = R(I)	•
R (I) = M C (J) = M L (M) = 0 T (M) = 0 ND(I) = NJ (I+1) ND(J) = NJ (J+1) GO TO 20 C STORE DIAGDNAL STOP R S00 4900 R S00 4900 R S00 5000 R S00 5000 R S00 5000 R S00 500 R S00 500 R S00 500 R S00 5600 R R S00 5600 R		C(4) = C(J)	
C (J) = M L (M) = 0 T (M) = 0 ND(I) = NJ (I+1) ND(J) = NJ (J+1) GO TO 20 C STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL RS00 5000 RS00 5000 RS00 5000 RS00 5000 RETJRN C ERROR-INSUFFICIENT STORAGE STOP RS00 5000 RS00 5000		R(I) = M	
L (M) = 0 T (M) = 0 ND(I) = N) (I+1) ND(J) = N) (J+1) GO TO 20 C STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL STORE JIAGONAL RS00 5000 RS00 5000 RS00 5100 RS00 5100 RS00 5100 RS00 5100 RS00 5000 RS00 5000		C(J) = M	
T (4) = 0 ND(I) =N) (I+1) ND(J) =N) (J+1) GO TO 20 C STORE DIAGDNAL STORE DIAGDNAL STORE DIAGDNAL ND(I) = V ND(I) = V ND(I) =N) (I+1) 20 CONTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 RETJRN C ERROR-INSUFFICIENT STORAGE NRITE (6,80) 80 FORMAT (2140INSJFFICIENT STORAGE) STOP RS005900 RS005900		L(M) = 0	
ND(I) =NJ(I+1) ND(J) =NJ(J+1) GO TO 20 C STORE DIAGDNAL STORE DIAGDNAL STORE DIAGDNAL RS00 4900 RS00 5000 RS00 5000 RS00 5000 RS00 5100 RS00 5100 RS00 5100 RS00 5400 RS00 5400 RS00 5500 RETJRN C ERROR-INSUFFICIENT STORAGE STOP RS00 5800 RS00 5900 RS00 5900 RS00 5900		T(4) = 0	
NU(J)=NJ(J+1)       RS004900         GO TO 20       RS005000         C STORE JIAGDNAL       RS005000         50       B(I) = V       RS005000         ND(I)=N)(I+1)       RS005100         20 CONTINUE       25 CONTINUE         25 CONTINUE       RS005400         C END OF MATRIX ELEMENTS       RS005400         GO RETJRN       RS005500         C ERROR-INSUFFICIENT STORAGE       RS005600         70       WRITE (6,80)         80       FORMAT (2140INSJFFICIENT STORAGE)         STOP       RS005900		ND(I) = NJ(I+1)	
C0 TO 20       RS004900         C STORE DIAGONAL       RS005000         50       B(I) = V       RS005100         ND(I) = N) (I+1)       20 CONTINUE         20 CONTINUE       25 CONTINUE         25 CONTINUE       8005400         C END OF MATRIX ELEMENTS       8005400         C ERROR-INSUFFICIENT STORAGE       RS005500         70       WRITE (6,80)         80       FORMAT (2140INSJFFICIENT STORAGE)         STOP       RS005900		ND(J) = NJ(J+1)	
C       STURE JIAGJNAL       RS00500         50       B(I) = V       RS005100         ND(I) = N)(I+1)       20 CONTINUE       25 CONTINUE         25       CONTINUE       25 CONTINUE         C       END OF MATRIX ELEMENTS       RS005400         G0       RETJRN       RS005500         C       ERROR-INSUFFICIENT STORAGE       RS005600         70       WRITE (6,80)       RS005800         80       FORMAT (2140INSJFFICIENT STORAGE)       RS005800         STOP       RS005900	~		RS004900
SU       B(I) = V       RS005100         ND(I) = N) (I+1)       20 CONTINUE       25 CONTINUE         25 CONTINUE       C       RS005400         C       END OF MATRIX ELEMENTS       RS005400         C       RETJRN       RS005500         C       ERROR-INSUFFICIENT STORAGE       RS005600         70       WRITE (6,80)       RS005800         80       FORMAT (2140INSJFFICIENT STORAGE)       RS005800         STOP       RS005900       RS005900	5	SIDRE JIAGJNAL	KS UU 5UU U
NO(1) = N)(1+1)         20 CONTINUE         25 CONTINUE         C END OF MATRIX ELEMENTS         60 RETJRN         C ERROR-INSUFFICIENT STORAGE         70 WRITE (6,80)         80 FORMAT (2140INSJFFICIENT STORAGE)         STOP	50		K2002100
25 CONTINUE 25 CONTINUE C END OF MATRIX ELEMENTS 60 RETJRN C C ERROR-INSUFFICIENT STORAGE 70 WRITE (6,80) 80 FORMAT (2140INSJFFICIENT STORAGE) STO <sup>2</sup> RS005800 RS005900			
C END OF MATRIX ELEMENTS       R\$005400         G0 RETJRN       R\$005500         C       C         C       ERROR-INSUFFICIENT STORAGE         70 WRITE (6,80)       R\$005600         80 FORMAT (2140INSJFFICIENT STORAGE)       R\$005800         STO?       R\$005900			
60         RETJRN         RS005400           60         RS005500         RS005500           C         C         RS005600           70         WRITE (6,80)         RS005600           80         FORMAT (2140INSJFFICIENT STORAGE)         RS005800           STOP         RS005900         RS005900	c	29 GUNIINUL END GE MATSTY EIEMENTS	
C         RS005500           C         RS005500           C         RS005600           C         RS005600           70         WRITE (6,80)           80         FORMAT (2140INSJFFICIENT STORAGE)           STO?         RS005900	60		RS 0 0 5400
C     ERROR-INSUFFICIENT STORAGE     RS005600       70     WRITE (6,80)     80       80     FORMAT (2140INSUFFICIENT STORAGE)     RS005800       STO?     RS005900	c.		RS005500
70         WRITE (6,80)         RS005800           80         FORMAT (2140INSJFFICIENT STORAGE)         RS005800           STO*         RS005900	č	EPPOR-TNSHEFTCTENT STOPAGE	RS 0 0 560 0
80 FORMAT (2140 INSJFFICIENT STORAGE) RS00 5800 STO <sup>2</sup> RS00 5900	70		
ST0> RS005900	80	FORMAT (2140 INSUFFICIENT STORAGE)	RS00 5800
		STO?	RS005900
END RS006000		E ND	RS006000

	SUBROUTINE LU(IPONE,UP)	
# #4	• * * * * * * * * * * * * * * * * * * *	* * *
#		+
¥	THE LU SUBROJTINE DECOMPOSES THE SYMMETRIC MATRIX.	♣
#		+
***	• * * * * * * * * * * * * * * * * * * *	* * *
С		
č	THE ROJTINE DECOMPOSES THE SYMMETRIC	LUSU 0700
č	MATRIX WHICH HAD BEEN ESTABLISTED BY	LUS0 0800
č.	THE READ ROUTINE. IF TRONE IS ZERO.	LUS0 090 0
č	THEN THE ROUTINE SELECTS THE PLYOTS	LUS01000
č	ACCORDING TO THE PLYOTING STRATEGY.	LUS01100
ř	OTHERWISE IT ASSUMES THAT THE DRDER	LUS01200
č	OF PIVATS IS PROVIDED IN THE ARRAY IP.	LUS01300
č		LUS01400
ř	THE PLITTNE USES TAG T FOR MARKING THE	LUS01500
č	THE AUDENT DECOMPOSED PART OF THE MATRIX.	LUS0 1600
ř	TT USES TAS I SUCH THAT THE POW OF A	LUS01700
č	The data first the second first first of the first of the second f	111501800
č	$\begin{array}{c} \textbf{P}  \textbf{P}  \textbf{V}  $	10501900
č	Connected and the rest connected approximate $T_{\rm eff}$ and $T_{\rm eff}$	10502000
Č	SET OF RIGHT CONNECTED AND ATTA ING E - De	20002000
č	STODAGE ASSTRANT NO SYMMETRIC DECOMPOSITION	5800 0200
č	STORAGE ASSIGNMENT FOR STHILLED OF NON-ZERY CONFERINTS.	58000300
5	THE THE MAXIMUM ALLOWED REFEACE THE NATETY.	58000400
ι.	NA - THE HARINGH ALLOWED SIZE OF THE HARINAA	58000500
~		SB000700
	ARKATS UP SIZE TO THE FOR FOR FOR LINKAGE	59000700
C	R - INIEGEX ARRAY JSED FUR KUW LINNAGE	58000000
C	C = INIEGEX ARXAT JSED FOR COLUMN LINAGE	50000500
C	L - INIEGER ARRAY USED FOR TAGGING ( ) OR 1 /	50001000
C	T - INFEGER ARRAY JSEU FUR TAGGING ( U UR I )	50001100
С	B - FL, PI, ARRAY JUNIAINING THE VALUES OF THE COEFFICIENTS	20001500
		59604500
_		50001500
C	ARRAYS OF SIZE N -	50001000
С	IP - INTEGER ARRAY CONTAINS THE SEQUENCE OF PIOUS	50001/00
C	ND - INTEGER ARRAY CONTAINS THE NUMBER OF ELEMENTS	58001000
С	IN A ROW OF THE UNDECOMPOSED PART OF THE MATRIX	28001900
C	IH1, IH2 - ARE TEMPORARY INTEGER ARRAYS, ONE OF THEM	58002000
С	MAY BE EQUIVALENCED TO IP	28002100
	COM4ON / ARRAYN/ IP(500), ND (500), IH 1 (500), 1H2 (500)	000000000
С	INDIVIJUAL DATA -	58902300
С	M - NUABER OF NONZERO ELEMENTS	58002400
С	N - SIZE OF THE MATRIX	SB002500
С	UP - PIVOT SELECTION PARAMETER	2000 2000
С	CT1 - MAXIMUM ELEMENT IN THE ORIGINAL MATRIX	28005100
С	CT2 - MAXIMUM ELEMENT ENCOUNTERED DURING DECOMPOSITION	28005900
	COMMON /DATA/ M, N, CT1, CT2	
С		
	LOGICAL SHP	LUSU2100
С		
	CT2 = CT1	LUSU 2300
	SW <sup>9</sup> = .TRUE.	
	IF (IPONE.NE.O) SWP=.FALSE.	LUSU 2500
С	LOOP FOR THE N PIVOTS	LUS02600

.

DO 300 T=1.N	LUS0 2700
	LUS0 2800
1 + (3 + 7) = 0 + 10 = 0 = 0 = 21	LUS02900
SAVE THE DE VOIS IN NO TE THEY HERE GIVEN	LUS-3300
DO IN MELAN	LUS0 3100
I NULT IL NET CIVEN PIVIT	LUS0 3300
	1.056.3600
	14503700
C PIANIS WERE NOT GIVEN, THOS FIND ONL	
30 UMAR = J.	1 1150 3900
IF (UPLE:0.) GU TU IIU	10504000
C FIND HAXINGLELEMENT IN THE AVAILABLE DIRGONALS	14504100
	2030 4100
	111504300
	10504600
10 0 CONTINUE	111504500
	111504600
C NOW FIND AN AVAILABLE JIAGUNAL WHUSE VALUE IS	
C GREATER THAN DHAX AND TAS MINUHUM NUMBER OF	10504100
C ELEMENTS IN ITS ROW	
110 IX = C	
DO 130 J=1,N	2030 9100
$IF(T(J) \cdot EQ \cdot 1 \cdot DR \cdot B(J) \cdot EQ \cdot J \cdot)$	14505700
IF (ABS(B(J)).LT.DMAX)	
1 G) TO 130	
IF (IX.=Q.J) GO 10 12J	2030 9800
IF (ND(J)-NY) 120,119,130	11150 5000
119 IF(ABS(3(J)) • LE• XM) = 010 130	
120  IX = J	
XM = ABS(B(J))	20300100
NY=ND (J)	1115067.10
13J CONFINUE	111576430
IF ((IX.EQ.U).02.	
1 (XM.LT.1.E-20) GJ TO 900	20308900
	11150 6700
C NOW THE PLUOT IS IN IX, SET ITS THE T	11506840
135  K = 1	20300000
	11150 7000
XMAX = 5 (1X)	10507100
G LUUP IS COLLECT THE ELEMENTS IN THE KOW OF THE FIGHT	10507200
17 = 1X	10507300
C COLLESI THE LEFT CONNESTED ACOS	10507403
C WIN INCLK KIONI INUEN ANCOC	
140 1T=C(1T) TE (TV =0 TV) CO TO 170	14507600
TE (1 (TK) EO 4) CO TO 440 TE (1 (TK) EO 4) CO TO 440	
TL (F(TL)*C(*T) OO IO TAO	
1 (17)=1	
160 12=J(12)	LUSD BOOD
1 (12.61.N) 60 10 100	1150 81 70
$K = K + 1 \qquad \cdot $	L0200100

IH1(K)=IY	
I H2 ( K ) =[ Z	
IF(SWP) NO(IZ) = ND(IZ-1)	
GO TQ 140	LUS0 8500
170  IY = IX	LUS08600
C COLLECT THE RIGHT CONNECTED ARCS	LUSO 870 J
C WITH THEIR LEFT INDEX VALUES	LUS0 8800
180 IY=3(IY)	
IF (IY.EQ.IX) GD TO 195	LUS0 9000
IF (T(IY).EQ.1) GO TO 180	
IZ = IY	LUS0 920 0
190 IZ=R(IZ)	
IF (IZ.JT.N) GO TO 190	LUS09400
K = K+1	LUS09530
IH1(K)=IY	
I H2 ( K ) =[ Z	
IF (SWP) ND(IZ)=ND(IZ-1)	
L (IY) =1	
T(IY) = 1	
GO TO 190	LUS10100
195 IF (K.E	LUS10200
K1 = I+1	LUS1030ŭ
C LOOP ON THE COLLECTED ARCS	LUS104J0
DO 250 J1=K1,K	LUS105J0
IY=IH1(J1)	
IZ=IH2(J1)	
$\mathbf{Y} = \mathbf{B}(\mathbf{I}\mathbf{Y})$	LUS10800
C DIVIDE THE ELEMENT BY THE PIVOT AND	LUS10900
C MODIFY ITS CORRESPONDING DIAGONAL	LUS11000
B(IY) = B(IY) / XMAX	LUS11100
B(IZ) = B(IZ) - B(IY) + Y	LUS11200
IF (ABS(B(IY)).5T.CF2)	LUS113ú0
1 CT2 = ABS (B (I-Y))	LUS11400
IF (ABS(B(IZ)).GT.CT2)	LUS11500
1 CT2 = ABS(B([Z))	LUS116J0
IF (J1.EQ.K) GO TO 250	LUS11700
K2 = J1+1	LUS11800
C INSIDE LOOP FOR THE REST OF THE COLLECTED ARCS	LUS1190û
C TO MODIFY THE INTERSECTING ARCS	LUS12000
DO 240 J2=<2,K	LUS12100
IY1=IH1(J2)	
I Z1 = I H2( J2)	,
C FIND THE ARC W BETWEEN IZ-IZ1 DIRECTED FROM	LUS12400
C MIN(IZ,IZ1) TO MAX(IZ,IZ1) IF EXISTS	LUS12500
I1 = MINO(IZ, IZ1)	LUS12600
I2 = MAX 0 (IZ, IZ1)	LUS12700
L 1= ₹ ( I 1)	
L 2=3 ( I 2)	
200 IF (L1.EQ.I1.0R.L2.EQ.I2)	LUS1 3000
< GOTO 220	LUS1 3190
IF (L1.EQ.L2) GO TO 230	LUS1 3200
IF (L1.3T.L2) GD TO 210	LUS1 3300
L 2=3 ( L 2)	
GO TO 230	LUS1 3500
210 L1= ₹ (L1)	

GO TO 230	LUS1 3700
C IT DOES NOT EXIST, THUS CREATE ONE	1.1164.300.0
220 IF (M.E.Q. MX) GO TO 950	[HE4/000
M = M+1	C0314000
IF (SWP) NU(I1)=NU(I1+1) TE (CWP) ND(T2)=ND(T2+4)	
IF (SWP) NU(12)=NU(12+1)	LUS1 430.0
LL = m TE / L1 CE 16000 WPTTE (6.261) [ 1	
261 E 129 A T (1 Y . 3H AAA, T15)	
B(11) = 0.	LUS14400
R(L1) = R(11)	
C(L1) = C(I2)	
R(I1) =L1	
C(I2)=L1	
L(L1)=0	
T (L1) =)	11164 5400
C MODIFY THE VALUE OF THE INTERSECTING ARC	11151 5200
230  B(L1) = B(L1) - B(IY) + B(IY1)	10515200
$IF (ABS(B(L1)) \cdot 3 \cdot C(2))$	LUS1 5400
	LUS1 5500
	LUS15600
	LUS1 5700
REFLEN	LUS1 5800
C	LUS15900
C ++++ ERRORS ++++	LUS1 6000
C NUMERIJALLY SINGULAR	LUS16100
900 WRITE (5,910) I	LUS162JU
910 FORMAT (24HO NUMERICALLY SINGULAR AT, 15)	
STOP	LUS16500
C INSUFFICIENT SIDRAJE	LUS1 6600
950 WELLE (3,900) I SCA FORMAT (2,000 THE FETTENT STORAGE AT T5)	LUS1 6700
ADA LAKANI (CAMATNOTLITENI DIAKWOR WINTA)	LUS16800
	LUS16900

	5	SUBROUTI	NE S	OL VE	(W)											SSOO	0100
**	****	* * * * * * * *	* ** *	****	****	* * * *	* * * * *	* * * *	** **	****	****	* * * *	* * * * *	* ** **	******	<b>#</b>	
*					_				_							¥	
*	-	THE S	OLVE	SUB	ROUT	INE	SOLVI	EST	HE T	HES	YMME	TRIC	LINE	AR SY	STEM	<b>+</b> 	
÷.	F	FOR THE	GIVE	NRI	GHT	HAND	SIDE	E W.	THE	SO	UTIO	NIS	RETU	RNED		<b>+</b>	
÷.		THROUGH	W. T	HEJ	OEF-	ICIE	NT M/	AIRL	X MU	IST H		BEFN	DECO	MUSED	PRIOR	-	
Ţ		IO GALLI	NG I	H12	SUBK		NE.		LUKK	ESPL	INUIN	6 35		E UF	PIVUIS	т в	
, in the second		12 12201	EUI	0 8:	ΙN	INE .	1P A1	<b>CICAT</b>	•							¥	
**				****	****	****	****		** **	****		****	* * * * * *	*****	*******	#	
Ċ.	1 000		ME I	ONED	TDT			T2V 2	EM							30.22	140.
č	2001	- 13 301	*	UNER	1 1 1 4		-	3131								5300	1410
č	STO	205 205		FNT	FOR	SYMM	FTRT	C DE	CO MP	OST	TON					SBOO	02ù0
č	MX -	- THE MA	X THU	MAL	LOWE	D NU	MBER	OF	NO N-	ZER	COE	FFIC	IENTS			SBOG	ú300
č	NX -	- THE MA	XIMU	M AL	LOWE	O SI	ZE OF	F TH	E MA	TRIX				•		SBOO	ú400
	Ċ	CONNON /	DIM/	HX.	NX											S800	0510
С	ARR	AYS OF S	I ZE	M -												SB00	0700
С	R -	INTEGER	ARR	AY J	SED	FOR	KOW I	LINK	AG E							SB 0 0	0830
С	с -	INTE GER	AR	AY U	SED	FOR	COLUI	MN L	IN KA	GE						SBOÙ	0660
С	L -	INTEGER	ARR	AY J	SED	FOR	TAGSI	ING	( 0	OR 1	. )					SB 0 0	1000
C	Τ-	INTEGER	ARR	AY J	SED	FOR	T A GG 🛛	ING	())	OR 1	. )					S800	1100
С	8 -	FL. PT.	ARR	AY C	ONTA	ININ	G THI	E VA	LU ES	OF	THE	COEF	FICIE	NTS		SB 0 0	1200
С		THE AR	RAYS	R,C	,L A	ND T	CONT	TAIN	PAC	KED	INTE	GERS	•				
	(	CO440N /	ARRA	YMZ	R ( 50	00),	C ( 50 I	00),	L( 50	CO),	T(50	<b>(</b> ( )	₿ (2 0 u	00)			
	1	INTEGER	R,C,	Т												SB 0 0	1500
C	ARR	AYS OF S	IZE	N -												SBOO	1630
C	IP -	- INTEGE	RAR	RAY	CONT	AINS	THE	SEQ		EOF	PIV	OTS				SBOO	1700
C	ND -	- INTEGE	RAR	RAY	CONT	AINS	THE	NUM	BER	01 5	LEME	NIS				5800	1830
C		IN A RUA	OF	THE	UNU:	CUMP	USEU	PAR		1113	MAI	KTX KTX				5800	1900
C	IH1	, 1H2 -	ARE	ILMP	URAK	Y IN	16651	K AK	KA 15	, 01	IE UF	INE	n .			5000	2100
C	, , , , , , , , , , , , , , , , , , ,	HAY BE 1	ADDA	ALEN	UEU TO/F	101	r No (5)		TU 4 4	6000	TUO	(= ) 0				2000	2100
<b>^</b>	TNO				16(2	007,	10(2)	,	Tu T/	5007	9102	1910	<b>,</b> .			SBOD	23.10
č	M	NIN BED		0175	P0 -		NTS									SBOOD	2430
ř	N -	ST7F OF	THE	MAT	PTY		115									SBRR	2500
č	UP -	- PIVOT	SELE	L TT1	N PA	RAME	TFR									5800	2630
č	CT1	- MAXT	UMF	I FME	NT T	N TH	FOR	I G I N	AL M		X					SBOO	2700
č	CT2	- MAXIM	UME	LEME	NT E	NCOU	NTER	EDD	URIN	IG DE	COMP	OSIT	ION			SBOO	2800
Ŭ	0,5	COMMON /	DATA	/ M.	N.CT	1.CT	2										
С						-,											
-	(	DIMENSIO	NHO	1)												SSûu	0053
С	•																
	1	DO 100 E	= 2, N													SS 0 0	1500
	]	IX=IP(I)															
	1	IY=IX															
	20 1	IY=2(IY)															
		IF (IY.E	Q.IX	) GD	TO	40										SS 0 0	1900
		IF (L(IY	).EQ	(.0)	GO T	0 20											<b>.</b>
	1	IZ = IY														55 <b>0</b> û	2100
	30 ]	1 Z=3 ( I Z)	<b>.</b>	•	TA -	•										6646	
		18 (12.G 4778) -	∎N) ⊔/тч	60	10 3	U D / T 4	、									2200	2300
			W (T X	)-#(	1714	9111	,									2200	2500
		50 IU ZJ TV - TV														2200	2600
	<b>6</b> 0 1	1 1 - 1A TV=^ / TV\															2000
	20 1	TI-2 (TI)															

	IE (IY.FQ.TX) 60 TO 100	SS 00 2800
	IF (L(IY), FQ.1) GO TO 50	
	IZ = IY	SS 0 0 3 0 0 0
60	I Z=R ( I Z)	
	IF (IZ.3T.N) GO TO 50	SS003200
	W(IX) = W(IX)-W(IZ)+B(IY)	SS003300
	GO TO 50	SS 0 0 340 0
10 0	CONTINUE	SS00 3500
C LO	OP TO SOLVE DIAGONAL SYSTEM	SS00 3600
	DO 110 I=1,N	SS06 3700
11 0	W(I) = A(I) / B(I)	SS00 7800
C LO	OP TO SOLVE TRANSPOSED UPPER TRIANGULAR SYSTEM	SS003900
	DO 200 I=2,N	SS004000
	K = N+1+I	SS044100
	IX=IP(K)	
	IY = IX	SS00 4300
120	IY=R(IY)	
	IF (IY.EQ.IX) GO TO 140	55004500
	IF (L(IY).EQ.1) GO TO 120	
	IZ = IY	55004700
130	I Z=3 ( I Z)	<b>55 8</b> 6 4 <b>8</b> 3 6
	IF (IZ.5T.N) GO TO 130	55004900
	W(IX) = W(IX) + W(IZ) + B(IY)	55005000
	GO TO 120	55005100
140	IY = IX	5500 5200
150	IY=3 (IY)	5600E400
	IF (1Y+1Q+1X) GU TU 200	3340 9400
	IF (L(IT)+EQ+U) GU (U 190	SCOREGOD
		3300 9800
100		55005800
	$\frac{1}{1} \left( \frac{1}{2} \cdot 1$	55005000
	$\mathbf{u} (\mathbf{T} \mathbf{v}) \rightarrow \mathbf{u} (\mathbf{T} \mathbf{v}) = \mathbf{u} (\mathbf{T} \mathbf{v}) = \mathbf{u} (\mathbf{T} \mathbf{v})$	S3005500 SS005500
20.0		SS000000 SS06610.3
200		SS000100 SS006200
		\$\$006200
		33440044

.

# INITIAL DISTRIBUTION

Copies:

1	ONR 430/R. LUNDEGARD
1	ONR 432/L. BRAM
1	NRL/8441/J. HANSON
1	USNA DEPT MATH
1	USNA LIB
1	NAVPGSCOL/59Ci/G. CANTIN
1	NAVPGSCOL/MATH DEPT
1	NAVPGSCOL LIB
1	NROTC & NAVADMINU, MIT
1	NAVWARCOL
1	NOL 331/E. COHEN

Copies:

1	NAVSHIPYD I	PTSMH/LIB
1	NAVSHIPYD E	SN/LIB
٦	NAVSHIPYD I	HILA/LIB
1	NAVSHIPYD I	IORVA/LIB
1	NAVSHIPYD (	CHASN/LIB
1	NAVSHIPYD I	MARE/LIB
1	NAVSHIPYD (	BREM/LIB
1	NAVSHIPYD I	PEARL/LIB
1	AIR FORCE A	ERO RES LABS/P. NIKOLAI
12	DDC	

## CENTER DISTRIBUTION

Copies:

Copies:

1	0000	NELSON PERRY W
י 1	1725	CIEFODD I FDOV N 1D
1	1725	
1	1/25	JUNES REMBERT F JR
1	1725	RODERICK JOAN E
1	1725	ROTH PETER N
1	1745	NG CHRISTOPHER
1	1800	GLEISSNER GENE H
1	1802	SHANKS DANIEL
1	1802	FRENKIEL FRANCOIS N
1	1802	LUGT HANS J
1	1802	THEILHEIMER FEODOR
1	1805	CUTHILL ELIZABETH H
1	1830	ERNST HERBERT M
1	1830	CULPEPPER LINWOOD M
1	1830	WALTON THOMAS S
1	1840	
1	1842	EDDY ROBERT P
1	1842	MEALS L KENTON

1	1843	SCHOT JOANNA WOOD
1	1844	DHIR SURENDRA K
1	1844	EVERSTINE GORDON C
10	1844	GIGNAC DONALD A
1	1844	GOLDEN MICHAEL M
1	1844	HENDERSON FRANCIS M
1	1844	MATULA PETRO
1	1850	CORIN THOMAS
1	1860	SULIT ROBERT A
1	1880	CAMARA ABEL W
1	1890	GRAY GILBERT R
1	1890	TAYLOR NORA M
1	1892	GOOD SHARON E
1	1892	RUMSEY JUDITH J
1	1966	LIU YUAN-NING
30	5614	REPORTS DISTRIBUTION
1	5641	LIBRARY
1	5642	LIBRARY

,

• •



