INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-307

CSDSLE - A FORTRAN PACKAGE FOR THE SOLUTION OF SPARSE DECOMPOSED SYSTEMS OF LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

January 1983

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

CSDSLE - A FORTRAN PACKAGE FOR THE SOLUTION OF SPARSE DECOMPOSED SYSTEMS OF LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

Abstract

CSDSLE is a package of forty-eight subroutines for solving sparse linear equations for the iterative simulation of very large systems. The equations are assumed to be in decomposed form as required by the main subroutine. Zollenkopf's bi-factorization-type algorithm is used to represent the solution of different submatrices. Different options of the package, when utilized properly, may save a significant amount of computer time and memory compared with the standard sparse-matrix subroutines, including those which utilize simple decomposition or block decomposition. The package and documentation have been developed for the CDC 170/730 system with the NOS 1.4 level 552 operating system and the Fortran Extended (FTN) Version 4.8 compiler. The report includes 17 illustrative examples, including the use of mass storage, the implementation of changes in the coefficient matrix and changes in the right-hand side. Local area changes and their effects on the rest of the system are discussed. A comparison with the Harwell package MA28 not using decomposition is reported.

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, Warsaw, Poland.

I. INTRODUCTION

The main aim of this package is to serve as an efficient tool in the solution and optimization of extremely large electrical or electronic networks. However, it may be utilized to solve any very large scale problem, which requires the solution of linear equations. The method developed to achieve this goal considers two aspects of the problem. Firstly, the size of the problem makes necessary the use of decomposition to fit the requirements to the available computer memory. Secondly, changes in coefficients and/or right hand side values occurring in a local area may cause local effects only, therefore the solution outside this region remains essentially unchanged. Accordingly, the method operates with a decomposed system of equations, in which changes in the coefficient matrix can be introduced locally and the solution can be obtained for a specified local area. The other important feature of the method is that, because hierarchical decomposition is employed, the computations can be executed in a parallel processing mode, thus further reducing the time necessary to obtain a solution.

The package is written in Fortran IV for the CDC 170/730 system. At McMaster University it is available in the form of a library of binary relocatable subroutines, which is linked with the user's program by the appropriate call of the main subroutine in the package. The name of the library is CSDSLE. The library is available as a group indirect file under the charge RJWBAND. The sequence of NOS commands to use the package can be as follows:

/GET(CSDSLE/GR) - fetch the library,

/LIBRARY(CSDSLE) - indicate the library to the loader,

/FTN(...,GO) - compile, load and execute the program.

The user should prepare:

- the main program which defines parameters and calls the main subroutine of the package,
- random access file containing the data describing submatrices.

II. GENERAL DESCRIPTION

Flowgraph Representation

Matrix decomposition is based on its Coates signal-flow graph representation [1], in which a square matrix $\underset{\sim}{A} = [a_{ij}]_{nxn}$ is represented by a graph with n nodes and k edges, and k is the number of nonzero coefficients in $\underset{\sim}{A}$. A Coates graph edge which goes from node x_j to node x_i has a weight equal to a_{ij}.

Example 1

Coates graph of a coefficient matrix

$$\underset{\sim}{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \\ 6 & 0 & 7 \end{bmatrix}$$

is shown in Fig. 1.

Graph Decomposition

A signal flow graph is decomposed through its nodes into two sub-graphs (subnetworks). Each of these subgraphs can be decomposed further down to a sufficiently small size. This kind of graph decomposition is called hierarchical decomposition. The structure of hierarchical decomposition can be illustrated by a tree of decomposition. Nodes of

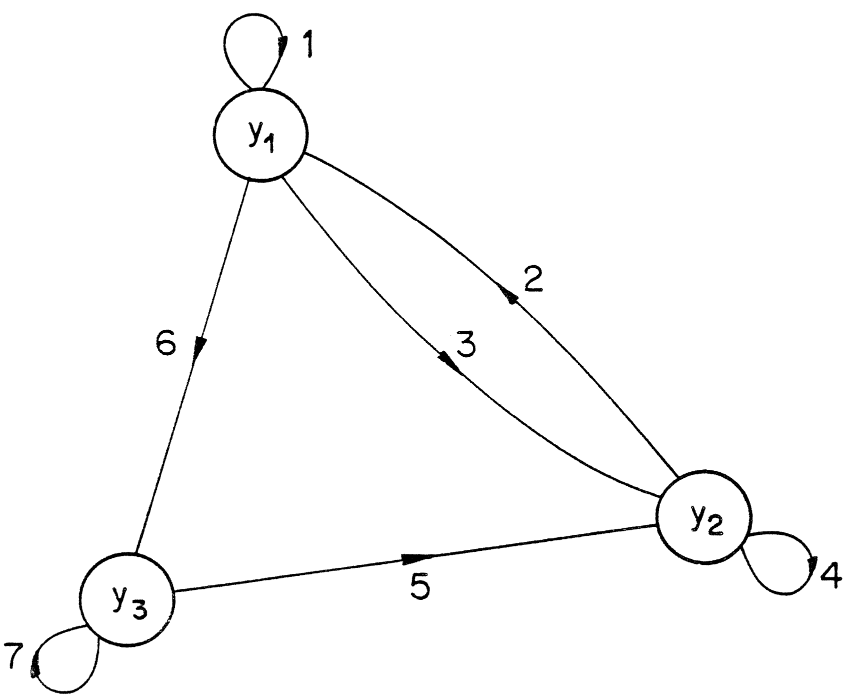


Fig. 1 Coates graph of the matrix $\underset{\sim}{A}$ of Example 1.

the tree correspond to subgraphs obtained on different levels of decomposition. If a subgraph G_j was obtained during decomposition of subgraph G_i, then there is an edge from the node corresponding to G_i to the node corresponding to G_j. Fig. 3 shows the tree of decomposition corresponding to Fig. 2.

In the decomposition tree we have one initial node - the one which is only the starting point of edges. Terminal nodes are those which are only the end points of edges. All nodes that are not terminal nodes are middle nodes. Subgraphs associated with terminal nodes are called proper blocks. We limit ourselves to bisection as the only graph partition so that every middle node has exactly two descendants. If m is the index of a middle subgraph then two of its descendants have indices 2m and 2m+1, respectively. This way of numbering the graphs makes the analysis of interconnections easier.

Matrix reordering

After all subgraphs have been numbered according to the structure of the decomposition tree, the nodes of a graph are renumbered consecutively in descending order starting from the partition nodes of the graph G_1, then the partition nodes of graph G_2 and G_3 up to the last partition and then the internal nodes of the proper blocks. After the renumbering, the numbers associated with the graph nodes are called original indices of the nodes. Such renumbering corresponds to reordering the coefficient matrix. An example for the matrix partitioned according to Fig. 2 is shown in Fig. 4.

In Fig. 4, only the shaded areas may contain nonzero coefficients. J_j denotes the index set corresponding to the partition nodes of graph

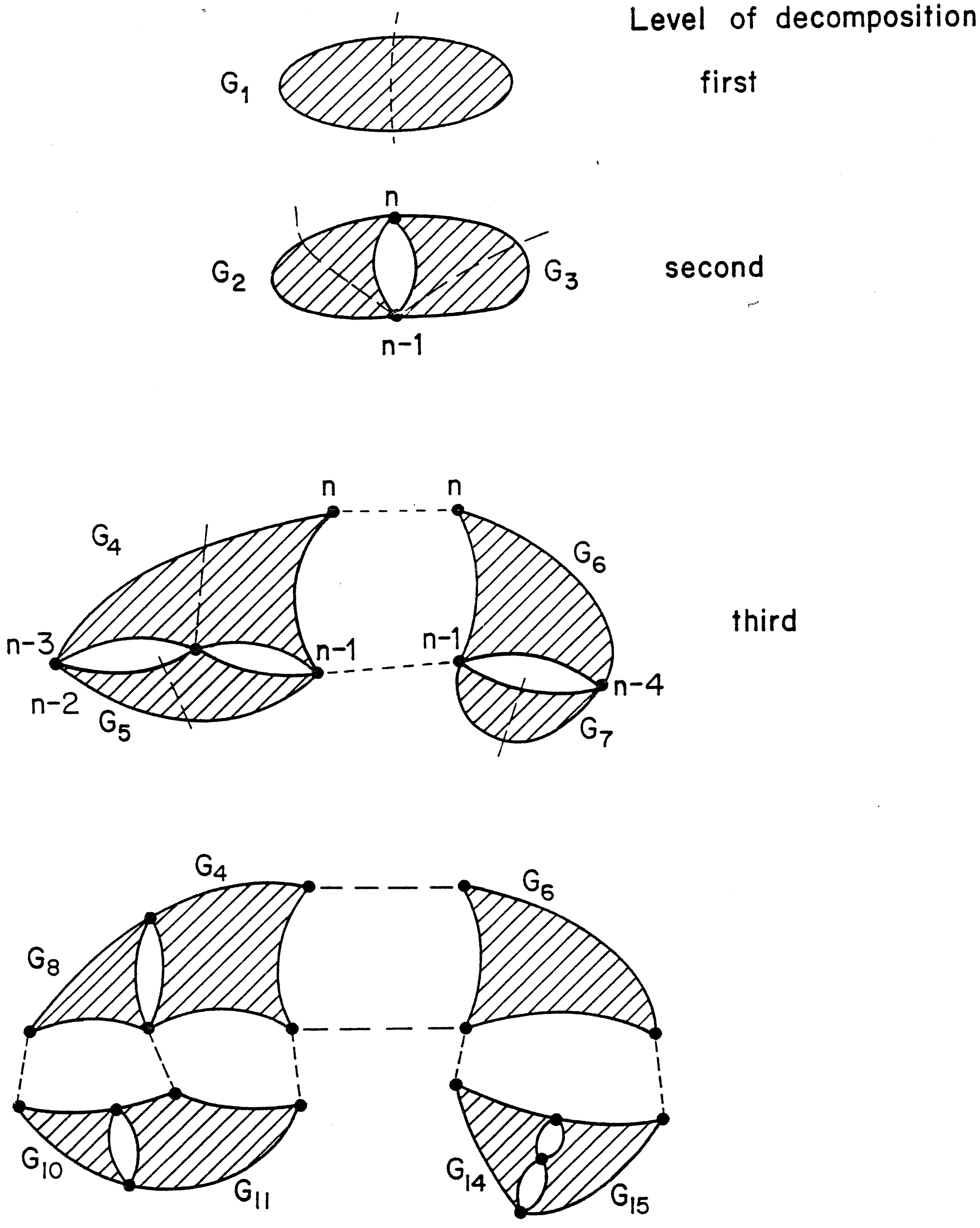


Fig. 2 Three level hierarchical decomposition.

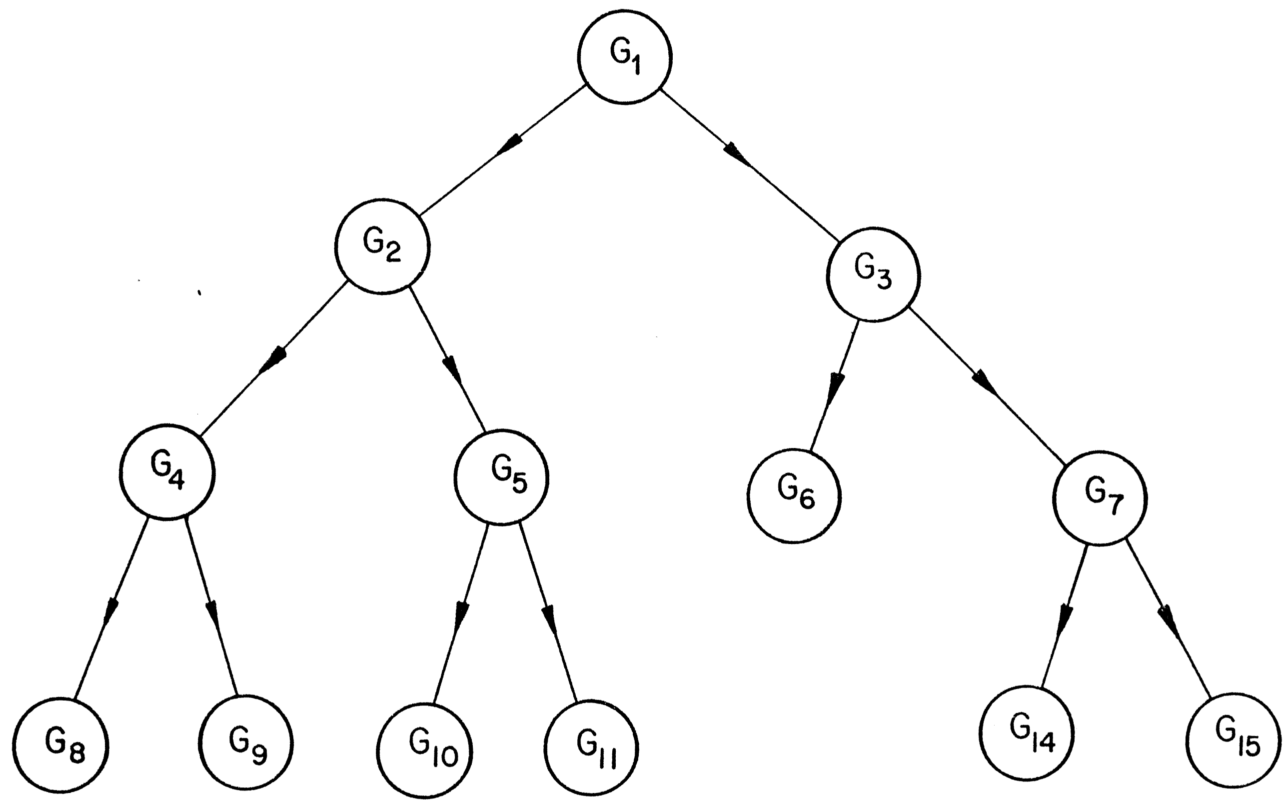


Fig. 3 Tree of decomposition for Fig. 2.

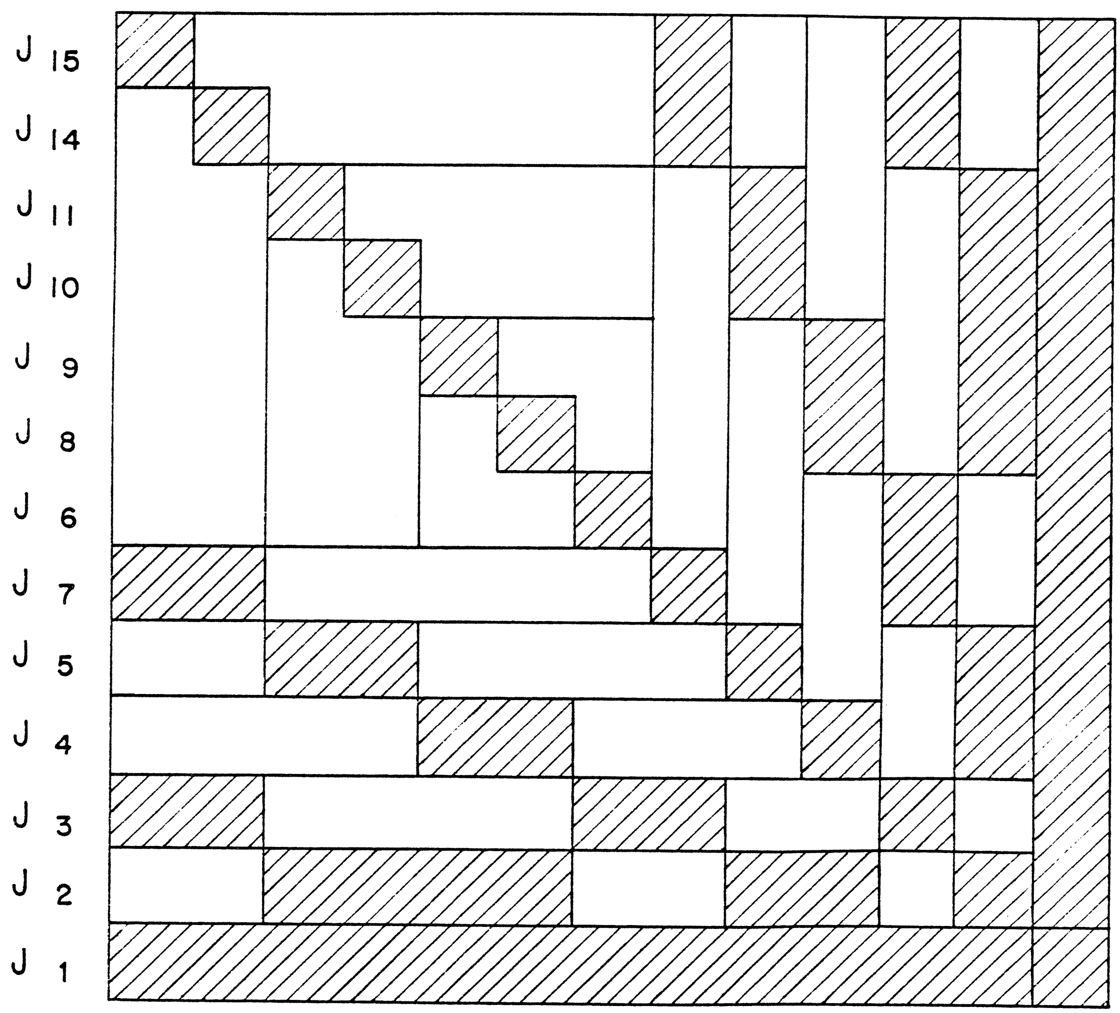


Fig. 4 Nonzero pattern of a reordered coefficient matrix.

G_j (if G_j was further partitioned) or the internal nodes for proper blocks.

Obviously, if some nodes were partition nodes of a graph from the higher level they will not be included into J_j. For example, for hierarchical decomposition of Fig. 2, $J_2 = \{n-2, n-3\}$, as the node (n-1) is a partition node of G_1.

There is a strict correspondence between the set of partition nodes of a middle graph and submatrices of the reordered coefficient matrix. These submatrices are called <u>interconnection matrices</u> as they represent interconnection of two subsystems.

<u>Example 2</u>

Consider a system of linear equations

$$\underset{\sim}{A}\, \underset{\sim}{y} = \underset{\sim}{b}$$

having its coefficient matrix and right-hand side vector equal to

$$\underset{\sim}{A} = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 4 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 6 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 7 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 8 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 9 \end{bmatrix}, \quad \underset{\sim}{b} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 9 \\ 1 \\ 0 \\ 0 \\ 2 \\ -1 \end{bmatrix}.$$

Coates graph G_1 of the coefficient matrix is shown in Fig. 5. G_1

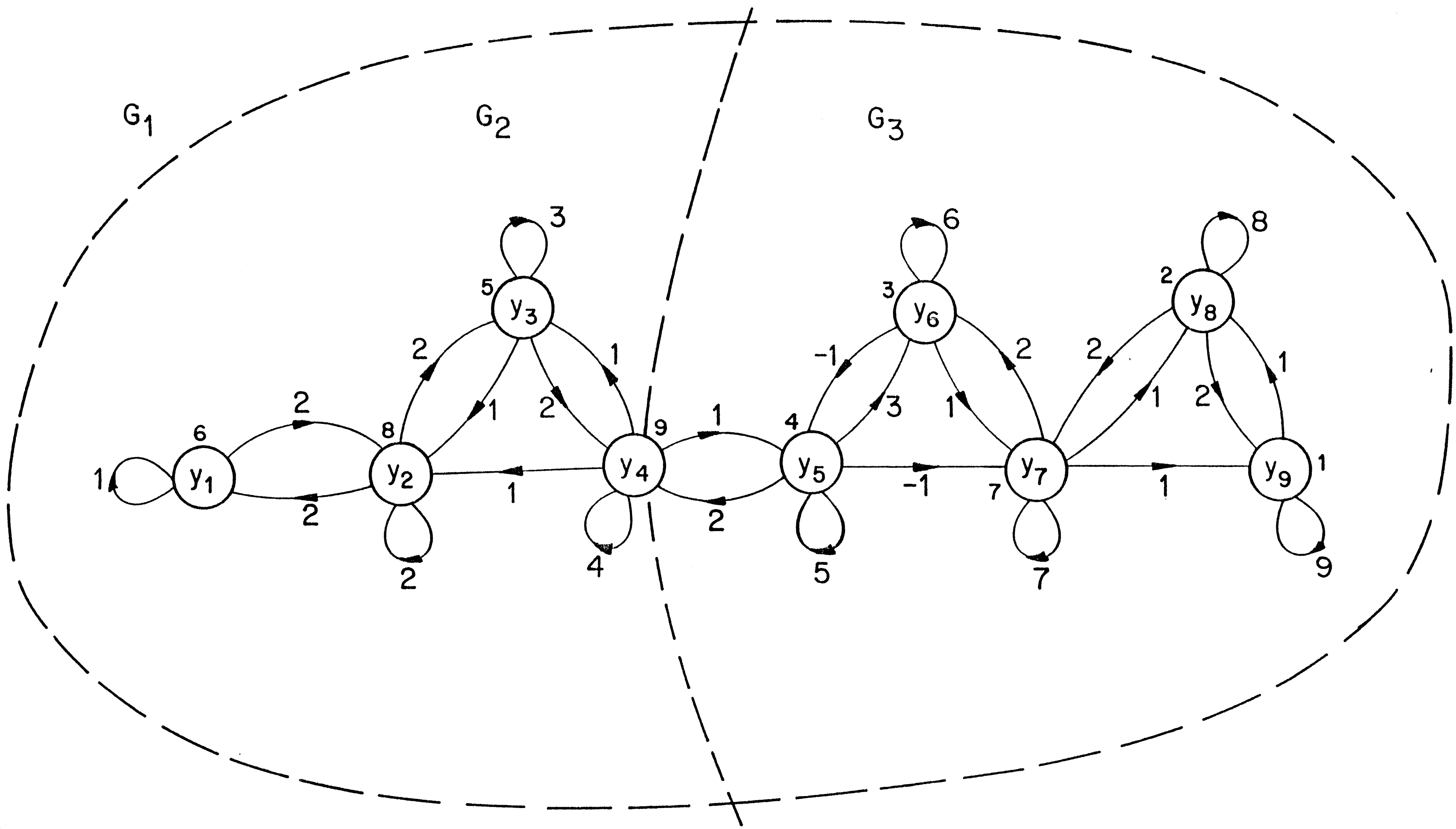


Fig. 5 Partitioned Coates graph G_1 for Example 2.

has been partitioned into G_2 and G_3 at the node y_4 and this node has been given the index 9, then G_2 has been partitioned into G_4 and G_5 at the node y_2 renumbered to 8 and G_3 has been partitioned into G_6 and G_7 at the node y_7 renumbered to 7. Finally, we obtain proper blocks G_4, G_5, G_6 and G_7, as shown in Fig. 6. In this case, the tree of decomposition is very simple, as shown in Fig. 7. The renumbered coefficient matrix has the nonzero pattern shown in Fig. 8. In fact, the renumbered matrix $\underset{\sim}{A}$ is actually more sparse than is shown in Fig. 8, namely

$$\underset{\sim}{A} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{bmatrix} 9 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 8 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 6 & 3 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & -1 & 5 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 2 & 1 & -1 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 4 \end{bmatrix} \end{array}$$

and the renumbered right hand side vector is

$$\underset{\sim}{b} = [-1 \quad 2 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 9]^T .$$

Sets of indices J_j according to the partition of signal-flow graph G_1 are as follows:

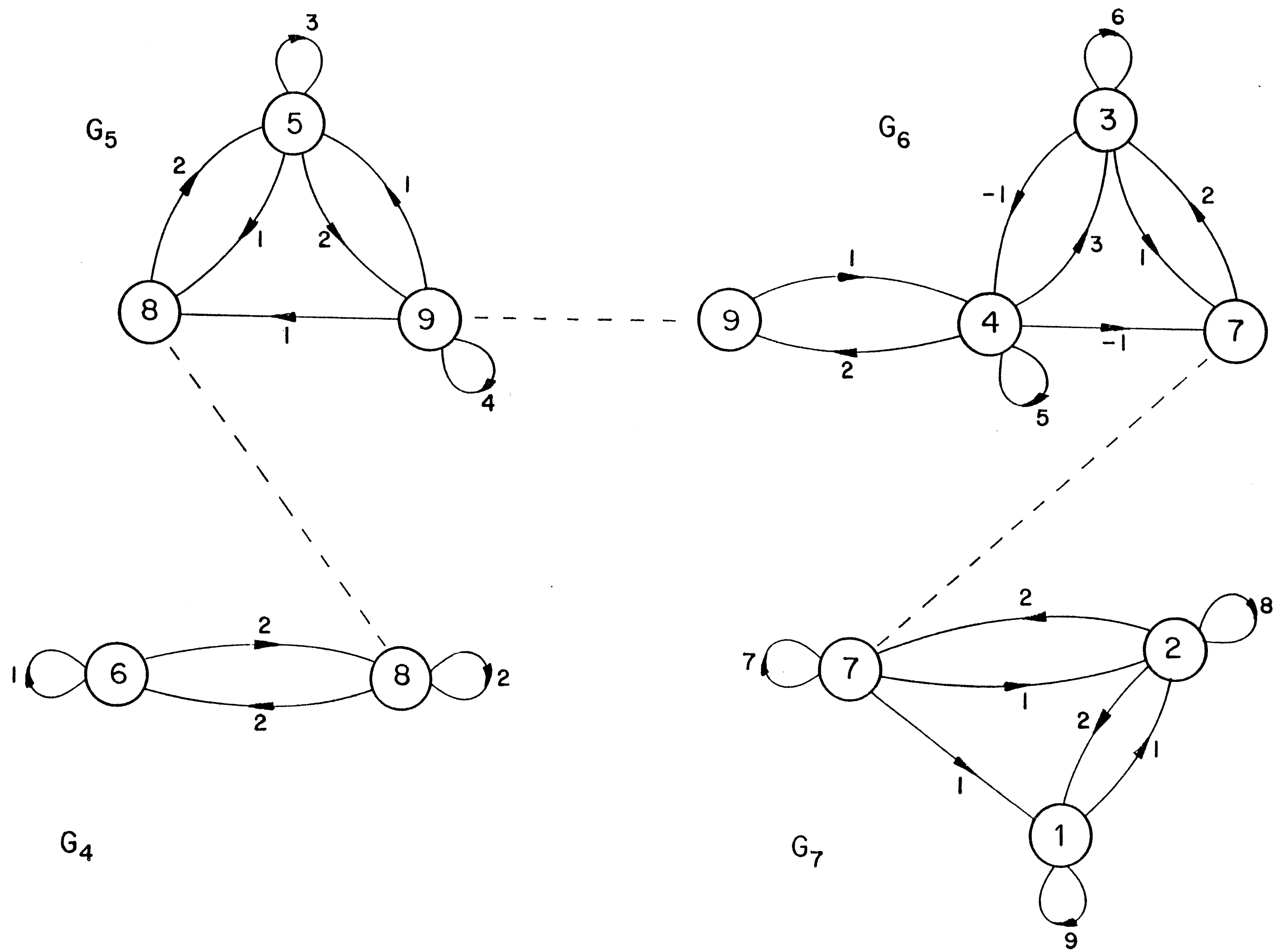


Fig. 6 Proper blocks for Example 2.

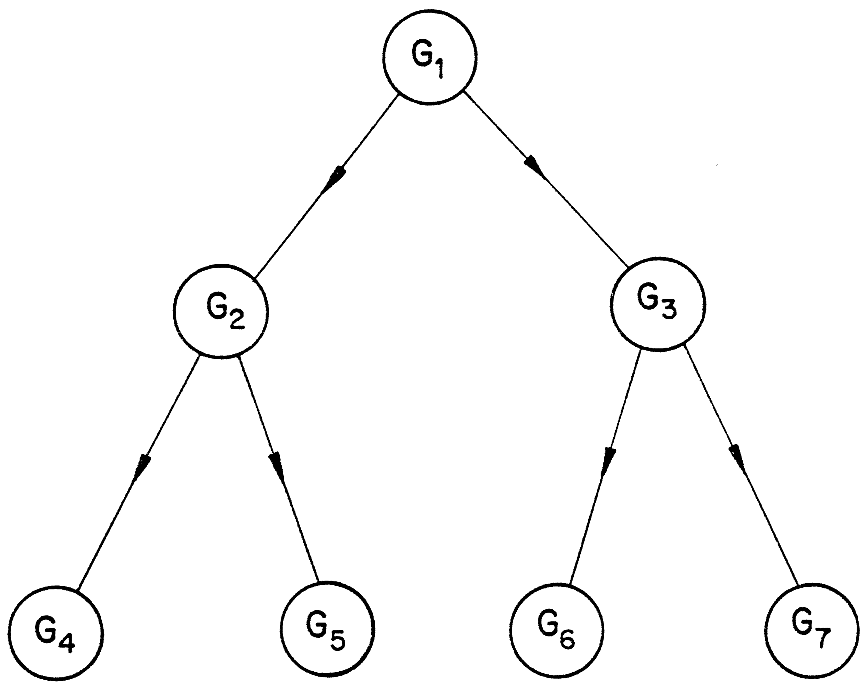


Fig. 7 Tree of decomposition for Example 2.

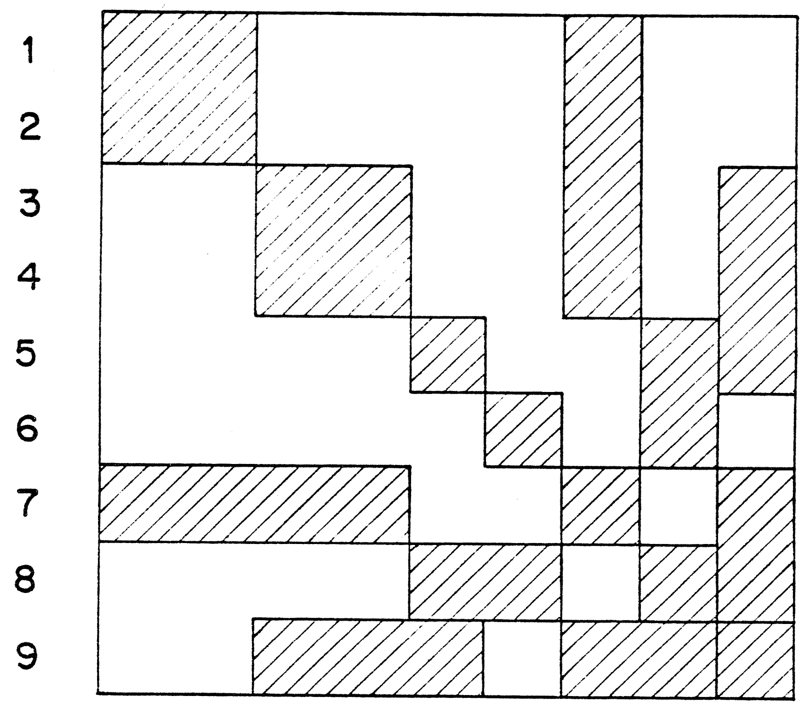


Fig. 8 Nonzero pattern of the coefficient matrix of Example 2.

$$J_1 = \{9\}, \quad J_2 = \{8\}, \quad J_3 = \{7\}, \quad J_4 = \{6\},$$
$$J_5 = \{5\}, \quad J_6 = \{3,4\}, \quad J_7 = \{1,2\}.$$

Hierarchical Decomposition in Bi-factorization Approach

The set of n linear equations

$$\underset{\sim}{A}\underset{\sim}{y} = \underset{\sim}{b} \tag{1}$$

can be solved effectively, if $\underset{\sim}{A}$ is a sparse matrix, by the use of the bi-factorization method [2]. The inverse of $\underset{\sim}{A}$ can be expressed by a multiple product of 2n factor matrices

$$\underset{\sim}{A}^{-1} = \underset{\sim}{R}^1 \underset{\sim}{R}^2 \ldots \underset{\sim}{R}^n \underset{\sim}{\bar{L}}^n \ldots \underset{\sim}{\bar{L}}^2 \underset{\sim}{\bar{L}}^1 , \tag{2}$$

where the left-hand factor matrices $\underset{\sim}{\bar{L}}^j$ are very sparse and differ from the unity matrix only in column j:

$$\underset{\sim}{\bar{L}}^j = \begin{bmatrix} 1 & & & & 0 & & & & \\ & \cdot & & & \cdot & & & & \\ & & \cdot & & \cdot & & & & \\ & & & \cdot & \cdot & & & & \\ & & & 1 & 0 & & & & \\ & & & & \bar{\ell}^j_{jj} & & & & \\ & & & & \bar{\ell}^j_{j+1,j} & 1 & & & \\ & & & & \cdot & & \cdot & & \\ & & & & \cdot & & & \cdot & \\ & & & & \cdot & & & & \cdot \\ & & & & \bar{\ell}^j_{n,j} & & & & 1 \end{bmatrix} . \tag{3}$$

The right-hand factor matrices $\underset{\sim}{R}^j$ are also very sparse and differ from the unity matrix only in row j:

$$\underset{\sim}{R}^j = \begin{bmatrix} 1 & & & & & & & & \\ & \cdot & & & & & & & \\ & & \cdot & & & & & & \\ 0 & \cdot & \cdot & 0 & 1 & r^j_{j,j+1} & \cdots & r^j_{j,n} \\ & & & & & 1 & & \\ & & & & & & \cdot & \\ & & & & & & & \cdot \\ & & & & & & & & 1 \end{bmatrix} . \tag{4}$$

The solution of the system (1) can be obtained by multiplying the product form of the inverse (2) by the right-hand side vector

$$\underset{\sim}{y} = \underset{\sim}{R}^1 \underset{\sim}{R}^2 \ldots \underset{\sim}{R}^n \underset{\sim}{\overline{L}}^n \ldots \underset{\sim}{\overline{L}}^2 \underset{\sim}{\overline{L}}^1 \underset{\sim}{b} . \tag{5}$$

After $\underset{\sim}{\overline{L}}^1$ is multiplied by $\underset{\sim}{b}$, we obtain the updated vector $\underset{\sim}{c}^1 = \underset{\sim}{\overline{L}}^1 \underset{\sim}{b}$, then we calculate $\underset{\sim}{c}^2 = \underset{\sim}{\overline{L}}^2 \underset{\sim}{c}^1$ and after n steps $\underset{\sim}{c}^n = \underset{\sim}{\overline{L}}^n \underset{\sim}{c}^{n-1}$. Then the right-hand factor matrices are employed to obtain $\underset{\sim}{c}^{2n-j+1} = \underset{\sim}{R}^j \underset{\sim}{c}^{2n-j}$, $j=n, n-1, \ldots, 1$ and finally $\underset{\sim}{y} = \underset{\sim}{c}^{2n}$.

Following the technique proposed in [2] we can decompose each left-hand factor matrix $\underset{\sim}{\overline{L}}^j$ into a modified matrix $\underset{\sim}{L}^j$ and a diagonal matrix $\underset{\sim}{D}^i$, namely

$$\underset{\sim}{\overline{L}}^j = \underset{\sim}{L}^j \underset{\sim}{D}^j , \tag{6}$$

where the modified matrix $\underset{\sim}{L}^j$ differs from the unity matrix in only column j. This column is

$$[0 \ldots 0 \; 1 \; \ell^j_{j+1,j} \; \ell^j_{j+2,j} \ldots \ell^j_{nj}]^T ,$$

where

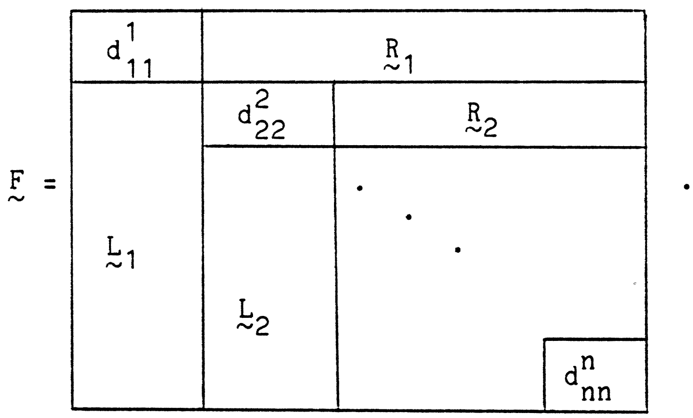
$$\ell^j_{ij} = \overline{\ell}^j_{ij} / \overline{\ell}^j_{jj} , \quad i = (j+1), \ldots, n$$

and the diagonal matrix $\underset{\sim}{D}^j$ differs from the unity matrix only in the ith diagonal term

$$d^j_{jj} = \bar{\ell}^j_{jj} \ .$$

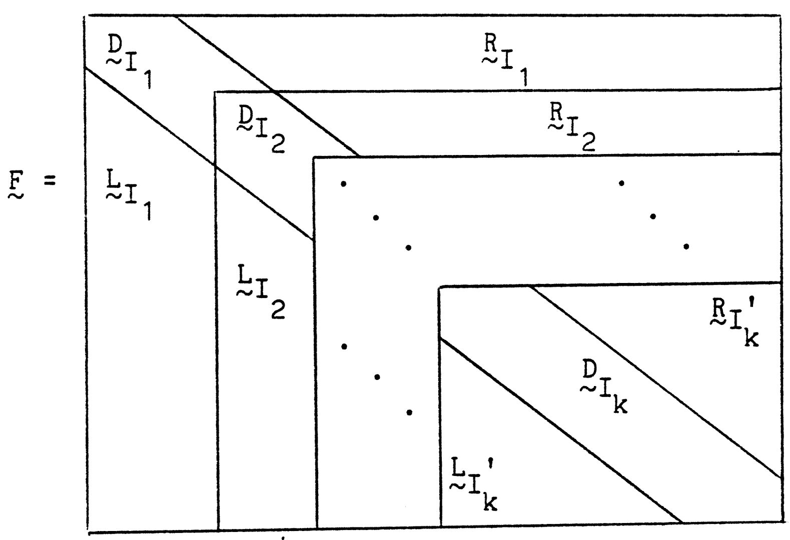
Let $\underset{\sim}{L}_j$ denote a column vector which contains elements of the jth column of $\underset{\sim}{L}^j$ starting from $\ell^j_{j+1,j}$, and let $\underset{\sim}{R}_j$ denote a row vector which contains elements of the jth row of $\underset{\sim}{R}^j$ starting from $r^j_{j,j+1}$.

A very common way of storing the factor matrices $\underset{\sim}{L}^j$ and $\underset{\sim}{R}^j$ is storing vectors $\underset{\sim}{L}_j$ and $\underset{\sim}{R}_j$ only together with the diagonal elements d^j_{jj} in the matrix form



(7)

By grouping together subsets of rows and columns, $\underset{\sim}{F}$ can be expressed in the form



(8)

where $I_j = \{m_j, m_j + 1, \ldots, \ell_j\}$ and $I'_j = I_j - \{\ell_j\}$.
Such groupings will help when the system is very large and $\underset{\sim}{L}_{I_j}$, $\underset{\sim}{D}_{I_j}$, $\underset{\sim}{R}_{I_j}$ can be calculated and stored separately. Then we can execute (5) step

after step, each time taking a group of matrices $\mathbf{L}^j$ and $\mathbf{D}^j$ or $\mathbf{R}^j$. This will save computer storage and allows us to analyse very large systems. If the system is extremely large, even the updated vectors $\mathbf{c}^j$ may be too big to be stored as a whole. In this case, we should partition the updated vectors $\mathbf{c}^j$ and groups of rows of $\mathbf{L}_{I_j}$ and columns of $\mathbf{R}_{I_j}$. This can be done very efficiently while a system is decomposed into blocks according to its topological signal-flow graph representation.

Let matrix $\mathbf{F}$ be written in the form

$$\mathbf{F} = \begin{bmatrix} \mathbf{L}_{I_1'I_1'} \;\; \mathbf{D}_{I_1} \;\; \mathbf{R}_{I_1'I_1'} & \mathbf{R}_{I_1I_2} & \cdots & \mathbf{R}_{I_1I_k} \\ \mathbf{L}_{I_2I_1} & \mathbf{L}_{I_2'I_2'} \;\; \mathbf{D}_{I_2} \;\; \mathbf{R}_{I_2'I_2'} & \cdots & \mathbf{R}_{I_2I_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_{I_kI_1} & \mathbf{L}_{I_kI_2} & \cdots & \mathbf{L}_{I_k'I_k'} \;\; \mathbf{D}_{I_k} \;\; \mathbf{R}_{I_k'I_k'} \end{bmatrix}, \qquad (9)$$

where $\mathbf{L}_{I_iI_j}$ $(\mathbf{R}_{I_jI_i})$ denotes a submatrix of $\mathbf{L}_{I_j}$ $(\mathbf{R}_{I_j})$ which contains elements of rows (columns) I_i.

Let column s of $\mathbf{L}_{I_iI_j}$ $(s \in I_j)$ represent a matrix which differs from the unity matrix only in column s:

$$
\underset{\sim}{L}^{s}_{I_i} = \left[\begin{array}{cccccccc}
1 & & & 0 & & & & \\
 & \ddots & & \vdots & & & & \\
 & & & 1 & & & & \\
 & & & 0 & 1 & & & \\
\hdashline
 & & & \ell^{s}_{m_i s} & & 1 & & \\
 & & & \vdots & & & \ddots & \\
 & & & \ell^{s}_{\ell_i s} & & & & 1 \\
\hdashline
 & & & 0 & & & & & 1
\end{array}\right] \begin{array}{l} \\ \\ \\ \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} I_i \\ \\ \end{array} , \tag{10}
$$

where $m_i, m_{i+1}, \ldots, \ell_i \in I_i$, $m_i > s$.

Let row s of $\underset{\sim}{R}_{I_j I_i}$ $(s \in I_j)$ represent a matrix which differs from the unity matrix only in row s:

$$
\underset{\sim}{R}^{s}_{I_i} = \left[\begin{array}{cccc:ccc:c}
1 & & & & \multicolumn{3}{c}{\overbrace{\hphantom{r^s_{sm_i} \quad \cdots \quad r^s_{s\ell_i}}}^{I_i}} & \\
 & \ddots & & & & & & \\
 & & 1 \quad 0 & \cdot \quad 0 & r^{s}_{sm_i} & \cdots & r^{s}_{s\ell_i} & 0 \cdot 0 \\
 & & 1 & & & & & \\
 & & & \ddots & & & & \\
 & & & 1 & & & & \\
 & & & & 1 & & & \\
 & & & & & \ddots & & \\
 & & & & & 1 & & \\
 & & & & & & \ddots & \\
 & & & & & & & \ddots \quad 1
\end{array}\right] , \tag{11}
$$

where $m_i, m_{i+1}, \ldots, \ell_i \varepsilon I_i$, $m_i > s$. Then, $\underset{\sim}{F}$ will represent a set of

$f = [2(\overset{=}{I}_1 k + \overset{=}{I}_2(k-1) + \ldots + \overset{=}{I}_k - k) + n]$ factor matrices $\underset{\sim}{L}^s_{I_i}$, $\underset{\sim}{D}^s$, $\underset{\sim}{R}^s_{I_i}$, where $^=$ denotes the cardinality of the corresponding set. The nonzero pattern of matrix $\underset{\sim}{F}$, according to the way $\underset{\sim}{F}$ is obtained from $\underset{\sim}{A}$, will be the same as for $\underset{\sim}{A}$ (see Fig. 4), so only those $\underset{\sim}{L}_{I_i I_j}$ and $\underset{\sim}{R}_{I_j I_i}$ will be nonzero, which correspond to the graph having the set of nodes $J_m = I_j$, and to all of its descendants.

Example 3

The renumbered coefficient matrix from Example 2 has its factor matrices represented by matrix $\underset{\sim}{F}$ as follows:

$$\underset{\sim}{F} = \left[\begin{array}{ccccccccc} \frac{1}{9} & -\frac{2}{9} & & & & & -\frac{1}{9} & & \\ -\frac{1}{9} & \frac{9}{70} & & & & & -\frac{8}{70} & & \\ & & \frac{1}{6} & -\frac{1}{2} & & & -\frac{1}{3} & & \\ & & \frac{1}{6} & \frac{2}{11} & & & -\frac{2}{33} & & -\frac{2}{11} \\ & & & & \frac{1}{3} & & & -\frac{2}{3} & -\frac{1}{3} \\ & & & & & 1 & & -2 & \\ & -\frac{9}{35} & -\frac{1}{6} & \frac{3}{11} & & & .15316 & & -.04177 \\ & & & & -\frac{1}{3} & -2 & & -\frac{3}{8} & \frac{1}{4} \\ & & & -\frac{4}{11} & -\frac{2}{3} & & .01856 & -\frac{1}{2} & .3785 \end{array}\right] \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} I_1 \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \end{array} .$$

It can be seen that the nonzero pattern of $\underset{\sim}{F}$ is the same as that of the coefficient matrix shown in Fig. 8.

According to the partition scheme of matrix $\underset{\sim}{F}$ shown in (9) we have

$\underset{\sim}{L}_{I_1' I_1'} = [-\frac{1}{9}]$, $\underset{\sim}{L}_{I_5 I_1} = [0 \quad -\frac{9}{35}]$,

$$\underset{\sim}{L}_{I_2'I_2'} = [\tfrac{1}{6}] , \quad \underset{\sim}{L}_{I_5I_2} = [-\tfrac{1}{6} \quad \tfrac{3}{11}] ,$$

$$\underset{\sim}{L}_{I_7I_2} = [0 \quad -\tfrac{4}{11}] , \quad \underset{\sim}{L}_{I_6I_3} = [-\tfrac{1}{3}] ,$$

$$\underset{\sim}{L}_{I_7I_3} = [-\tfrac{2}{3}] , \quad \underset{\sim}{L}_{I_6I_4} = [-2] ,$$

$$\underset{\sim}{L}_{I_7I_5} = [.01856], \quad \underset{\sim}{L}_{I_7I_6} = [-\tfrac{1}{2}] ,$$

$$\underset{\sim}{D}_{I_1} = \begin{bmatrix} \frac{1}{9} & 0 \\ 0 & \frac{9}{70} \end{bmatrix}, \quad \underset{\sim}{D}_{I_2} = \begin{bmatrix} \frac{1}{6} & 0 \\ 0 & \frac{2}{11} \end{bmatrix},$$

$$\underset{\sim}{D}_{I_3} = [\tfrac{1}{3}], \quad \underset{\sim}{D}_{I_4} = [1], \quad \underset{\sim}{D}_{I_5} = [.15316] ,$$

$$\underset{\sim}{D}_{I_6} = [-\tfrac{3}{8}], \quad \underset{\sim}{D}_{I_7} = [.3785] ,$$

$$\underset{\sim}{R}_{I_1'I_1'} = [-\tfrac{2}{3}], \quad \underset{\sim}{R}_{I_1I_5} = \begin{bmatrix} -\frac{1}{9} \\ -\frac{8}{70} \end{bmatrix}, \quad \underset{\sim}{R}_{I_2'I_2'} = [-\tfrac{1}{2}] ,$$

$$\underset{\sim}{R}_{I_2I_5} = \begin{bmatrix} -\frac{1}{3} \\ -\frac{2}{33} \end{bmatrix}, \quad \underset{\sim}{R}_{I_2I_7} = \begin{bmatrix} 0 \\ -\frac{2}{11} \end{bmatrix} ,$$

$$\underset{\sim}{R}_{I_3I_6} = [-\tfrac{2}{3}], \quad \underset{\sim}{R}_{I_3I_7} = [-\tfrac{1}{3}] ,$$

$$\underset{\sim}{R}_{I_4I_6} = [-2], \quad \underset{\sim}{R}_{I_5I_7} = [-.04177] ,$$

$$\underset{\sim}{R}_{I_6I_7} = [\tfrac{1}{4}],$$

where $I_1 = J_7$, $I_2 = J_6$, $I_3 = J_5$, $I_4 = J_4$, $I_5 = J_3$, $I_6 = J_2$, and $I_7 = J_1$, where J_j is as defined in Example 2.

Let us define the multiple products:

$$\xi_{I_i I_j} \circ \underset{\sim}{x} \triangleq \begin{cases} \underset{\sim}{D}^{\ell_j} \, \underset{\sim}{L}_{I_i'}^{\ell_j - 1} \, \underset{\sim}{D}^{\ell_j - 1} \cdots \underset{\sim}{L}_{I_i'}^{m_j} \, \underset{\sim}{D}^{m_j} \, \underset{\sim}{x} & \text{for } j=i \\ \underset{\sim}{L}_{I_i}^{\ell_j} \cdots \underset{\sim}{L}_{I_i}^{m_j+1} \, \underset{\sim}{L}_{I_i}^{m_j} \, \underset{\sim}{x} & \text{for } j \neq i \end{cases} \tag{12}$$

and let

$$\Phi_{I_j I_i} \circ \underset{\sim}{x} \triangleq \begin{cases} R_{I_i'}^{m_j} \, R_{I_i'}^{m_j+1} \cdots R_{I_i'}^{\ell_j - 1} \, \underset{\sim}{x} & \text{for } j=i \\ \underset{\sim}{R}_{I_i}^{m_j} \, \underset{\sim}{R}_{I_i}^{m_j+1} \cdots \underset{\sim}{R}_{I_i}^{\ell_j} \, \underset{\sim}{x} & \text{for } j \neq 1 \end{cases} . \tag{13}$$

In a particular case, if

$$I_j = \{m_j\}$$

then

$$\xi_{I_j I_j} \circ \underset{\sim}{x} = \underset{\sim}{D}^{m_j} \underset{\sim}{x},$$

and

$$\Phi_{I_j I_j} \circ \underset{\sim}{x} = \underset{\sim}{x}.$$

<u>Lemma 1</u>

The inverse $\underset{\sim}{A}^{-1}$ (2) can be obtained as a multiple product of f factor matrices according to the formula

$$\begin{aligned} \underset{\sim}{A}^{-1} \underset{\sim}{x} = {} & \Phi_{I_1 I_1} \circ \Phi_{I_1 I_2} \circ \cdots \circ \Phi_{I_1 I_k} \circ \\ & \circ \Phi_{I_2 I_2} \circ \cdots \circ \Phi_{I_2 I_k} \circ \cdots \Phi_{I_k I_k} \circ \xi_{I_k I_k} \circ \\ & \cdots \circ \xi_{I_k I_2} \circ \cdots \circ \xi_{I_2 I_2} \circ \xi_{I_k I_1} \circ \\ & \cdots \circ \xi_{I_2 I_1} \circ \xi_{I_1 I_1} \circ \underset{\sim}{x} . \end{aligned} \tag{14}$$

Lemma 2

Multiple product

$$\underset{\sim}{z} = \xi_{I_k I_k} \circ \dots \circ \xi_{I_k I_2} \circ \dots \circ \xi_{I_2 I_2} \circ$$
$$\circ\, \xi_{I_k I_1} \circ \dots \circ \xi_{I_2 I_1} \circ \xi_{I_1 I_1} \circ \underset{\sim}{x} \tag{15}$$

can be obtained as a sum of vectors

$$\underset{\sim}{z} = \sum_{j=1}^{k} \underset{\sim}{z}_{I_j}, \tag{16}$$

where

$$\underset{\sim}{z}_{I_j} = \xi_{I_j I_j} \circ (\xi_{I_j I_1} \circ \underset{\sim}{z}_{I_1} + \xi_{I_j I_2} \circ \underset{\sim}{z}_{I_2} + \dots$$
$$+ \xi_{I_j I_{j-1}} \circ \underset{\sim}{z}_{I_{j-1}} + [\underset{\sim}{x}_{I_j}]) , \tag{17}$$

where $[\underset{\sim}{x}_{I_j}] = [0, \dots 0, \underset{\sim}{x}_{I_j}^T, 0, \dots, 0]^T$. For a particular case

$$\underset{\sim}{z}_{I_1} = \xi_{I_1 I_1} \circ [\underset{\sim}{x}_{I_1}] . \tag{18}$$

Example 4

Using LR factorization as in Example 3, and $\underset{\sim}{x}$ equal to renumbered right-hand side vector $\underset{\sim}{b}$ as in Example 2, vectors $\underset{\sim}{z}_{I_j}$ are as follows:

$$z_{I_1} = \begin{bmatrix} -\frac{1}{9} \\ \frac{19}{70} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad z_{I_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{2}{11} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad z_{I_3} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad z_{I_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$z_{I_5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.041374 \\ 0 \\ 0 \end{bmatrix}, \quad z_{I_6} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.375 \\ 0 \end{bmatrix}, \quad z_{I_7} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3.4569 \end{bmatrix}.$$

So, on the basis of (16), we have

$$z = [-\frac{1}{9} \quad \frac{19}{70} \quad 0 \quad \frac{2}{11} \quad 0 \quad 1 \quad -0.041374 \quad 0.375 \quad 3.4569]^T .$$

Assume that all the multiple products of $\xi_{I_j I_\ell} \circ z_\ell$ (j=1,..,k) in (17) have been previously calculated and stored. If now the coefficients in only one proper block have been altered (for example in G_{14} of Fig. 3) then according to the formula (17) and the nonzero pattern of Fig. 4, we must calculate new values of $z_{J_{14}}$, z_{J_7}, z_{J_3} and z_{J_1}, to obtain a new value of z. We have

$$\underset{\sim}{z}_{J_{14}} = \xi_{J_{14}J_{14}} \circ [\underset{\sim}{x}_{J_{14}}] \; , \tag{19}$$

$$\underset{\sim}{z}_{J_7} = \xi_{J_7J_7} \circ (\xi_{J_7J_{15}} \circ \underset{\sim}{z}_{J_{15}} + \xi_{J_7J_{14}} \circ \underset{\sim}{z}_{J_{14}} + [\underset{\sim}{x}_{J_7}]) \; , \tag{20}$$

$$\underset{\sim}{z}_{J_3} = \xi_{J_3J_3} \circ (\xi_{J_3J_{15}} \circ \underset{\sim}{z}_{J_{15}} + \xi_{J_3J_{14}} \circ \underset{\sim}{z}_{J_{14}} + \xi_{J_3J_6} \circ \underset{\sim}{z}_{J_6} + \xi_{J_3J_7} \circ \underset{\sim}{z}_{J_7} + [\underset{\sim}{x}_{J_3}]) \; , \tag{21}$$

$$\begin{aligned} \underset{\sim}{z}_{J_1} = \xi_{J_1J_1} \circ (&\xi_{J_1J_{15}} \circ \underset{\sim}{z}_{J_{15}} + \xi_{J_1J_{14}} \circ \underset{\sim}{z}_{J_{14}} + \xi_{J_1J_{11}} \circ \underset{\sim}{z}_{J_{11}} \\ &+ \xi_{J_1J_{10}} \circ \underset{\sim}{z}_{J_{10}} + \xi_{J_1J_9} \circ \underset{\sim}{z}_{J_9} + \xi_{J_1J_8} \circ \underset{\sim}{z}_{J_8} + \xi_{J_1J_6} \circ \underset{\sim}{z}_{J_6} \\ &+ \xi_{J_1J_7} \circ \underset{\sim}{z}_{J_7} + \xi_{J_1J_5} \circ \underset{\sim}{z}_{J_5} + \xi_{J_1J_4} \circ \underset{\sim}{z}_{J_4} + \xi_{J_1J_3} \circ \underset{\sim}{z}_{J_3} \\ &+ \xi_{J_1J_2} \circ \underset{\sim}{z}_{J_2} + [\underset{\sim}{x}_{J_1}]) \; . \end{aligned} \tag{22}$$

Observe that all the multiple products with variables $\underset{\sim}{z}_{J_{15}}$, $\underset{\sim}{z}_{J_{11}}$, $\underset{\sim}{z}_{J_{10}}$, $\underset{\sim}{z}_{J_9}$, $\underset{\sim}{z}_{J_8}$, $\underset{\sim}{z}_{J_6}$, $\underset{\sim}{z}_{J_5}$, $\underset{\sim}{z}_{J_4}$, $\underset{\sim}{z}_{J_2}$ are known and do not have to be recalculated.

In this simple example we have to calculate only 10 multiple products out of 41 to update $\underset{\sim}{z}$. Whenever a certain block has updated coefficients we must repeat the bifactorization for this block and all submatrices representing its interconnections according to the decomposition tree. What is even more important is that the analysis of only one proper block must be repeated, while multiple products corresponding to the substitute subgraphs are not so time-consuming.

Lemma 3

Multiple product

$$\underset{\sim}{y} = \Phi_{I_1 I_1} \circ \Phi_{I_1 I_2} \circ \dots \circ \Phi_{I_1 I_k} \circ$$

$$\circ \Phi_{I_2 I_2} \circ \dots \circ \Phi_{I_2 I_k} \circ \dots \Phi_{I_k I_k} \circ \underset{\sim}{z}$$

can be obtained as a sum of vectors

$$\underset{\sim}{y} = \sum_{j=1}^{k} \underset{\sim}{y}_{I_j} , \tag{23}$$

where

$$\underset{\sim}{y}_{I_j} = \Phi_{I_j I_j} \circ [\underset{\sim}{z}_{I_j}] + \Phi_{I_j I_{j+1}} \circ \underset{\sim}{y}_{I_{j+1}} + \dots + \Phi_{I_j I_k} \circ \underset{\sim}{y}_{I_k} , \tag{24}$$

where $[\underset{\sim}{z}_{I_j}] = [0, \dots, 0, \underset{\sim}{z}_{I_j}^T, 0, \dots, 0]^T$. For a particular case

$$\underset{\sim}{y}_{I_k} = \Phi_{I_k I_k} \circ [\underset{\sim}{z}_{I_k}] . \tag{25}$$

In a very similar way, we can show that if we are interested in the solution for any submatrix then only those multiple products $\Phi_{I_j I_\ell} \circ \underset{\sim}{z}_\ell$ must be calculated which correspond to the set of indices J_j, J_ℓ for submatrices from a given one to the top of the decomposition tree.

The main advantage of this type of partial analysis is a great reduction in computer time, which is proportional to the number of nodes in the subgraph representing the updated submatrix rather than to the number of all nodes, when the coefficient matrix is being partly updated. When our goal is to recalculate the solution in a particular subnetwork only, rather than in the whole network, again the computational time will be proportional to the number of updated unknowns and not to the total number of variables.

Example 5

On the basis of LR factorization from Example 3 and the vector $\underset{\sim}{z}$ obtained in Example 4, we obtain vectors $\underset{\sim}{y}_{I_j}$ as follows:

$$\underset{\sim}{y}_{I_7} = \underset{\sim}{z}_{I_7},\ y_{I_6} = \underset{\sim}{z}_{I_6} + \Phi_{I_6 I_7} \circ \underset{\sim}{z}_{I_7}$$

$$= [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1.2392\ \ 0]^T,$$

$$\underset{\sim}{y}_{I_5} = \underset{\sim}{z}_{I_5} + \Phi_{I_5 I_7} \circ \underset{\sim}{z}_{I_7} = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -0.18577\ \ 0\ \ 0\ ]^T,$$

$$\underset{\sim}{y}_{I_4} = \underset{\sim}{z}_{I_4} + \Phi_{I_4 I_6} \circ \underset{\sim}{z}_{I_6} = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1.4885\ \ 0\ \ 0\ \ 0]^T,$$

$$\underset{\sim}{y}_{I_3} = \underset{\sim}{z}_{I_3} + \Phi_{I_3 I_6} \circ \underset{\sim}{z}_{I_6} + \Phi_{I_3 I_7} \circ \underset{\sim}{z}_{I_7}$$

$$= [0\ \ 0\ \ 0\ \ 0\ \ -1.9785\ \ 0\ \ 0\ \ 0\ \ 0]^T,$$

$$\underset{\sim}{y}_{I_2} = \Phi_{I_2 I_2} \circ \underset{\sim}{z}_{I_2} + \Phi_{I_2 I_5} \circ \underset{\sim}{z}_5 + \Phi_{I_2 I_7} \circ \underset{\sim}{z}_7$$

$$= [0\ \ 0\ \ 0.27966\ \ -0.43546\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T,$$

$$\underset{\sim}{y}_{I_1} = \Phi_{I_1 I_1} \circ \underset{\sim}{z}_{I_1} + \Phi_{I_1 I_5} \circ \underset{\sim}{z}_{I_5}$$

$$= [-0.1555\ \ 0.29266\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T.$$

The final solution is

$$\underset{\sim}{y} = [-0.1555\ \ 0.29266\ \ 0.27966\ \ -0.43546\ \ -1.9785\ \ -1.4785$$
$$-0.18577\ \ 1.2392\ \ 3.4569]^T.$$

III. STRUCTURE OF THE PACKAGE

There are 2 different entries to the package and two corresponding main subroutines:

1. subroutine CSDSLE1 - standard entry for analysis of decomposed

systems of linear equations,

2. subroutine CSSLE1 - entry for analysis of systems of linear equations without decomposition [3].

A block diagram of the package is shown in Fig. 9. Checking the input parameters and subdivision of the working areas (defined by the user as variables of indicator vector IN) is performed in CSDSLE1. Subroutine ASDNSR organizes the partial analysis process for updated submatrices. Submatrices to be analyzed have to be stored in random access file No. 2 or in the working area CMPLXWA as described in Section V. ABSUB organizes and stores information about the hierarchical structure, CHEBSUB checks if a subgraph has been previously analyzed. STOREF and READF are used to store and read integer vector IN in and from the mass storage file, respectively. Different submatrices are read with the help of READSB and checked in CHEDEL if all diagonal elements exist in the submatrix. If certain nodes are common to two or more submatrices, then diagonal elements must be nonzero at least in one of them. If any diagonal element is not defined then the user is notified by the statement

ALL DIAGONAL ELEMENTS SHOULD BE DEFINED IN THE SUBMATRIX NO. nr

and execution of the main subroutine is stopped with IN(3)=1, where nr denotes the index of the corresponding subgraph as numbered at the tree of decomposition.

Subroutine ASUBSR organizes partial analysis of one subgraph and stores the results of partial analysis in the basic record BREC. If the user changes the nonzero pattern of a submatrix such that a new submatrix requires more space in the basic record then the message

MODIFIED SUBGRAPH NO. nr REQUIRES MORE STORAGE THAN THE PREVIOUS ONE
REPEAT ANALYSIS FOR ALL SUBGRAPHS

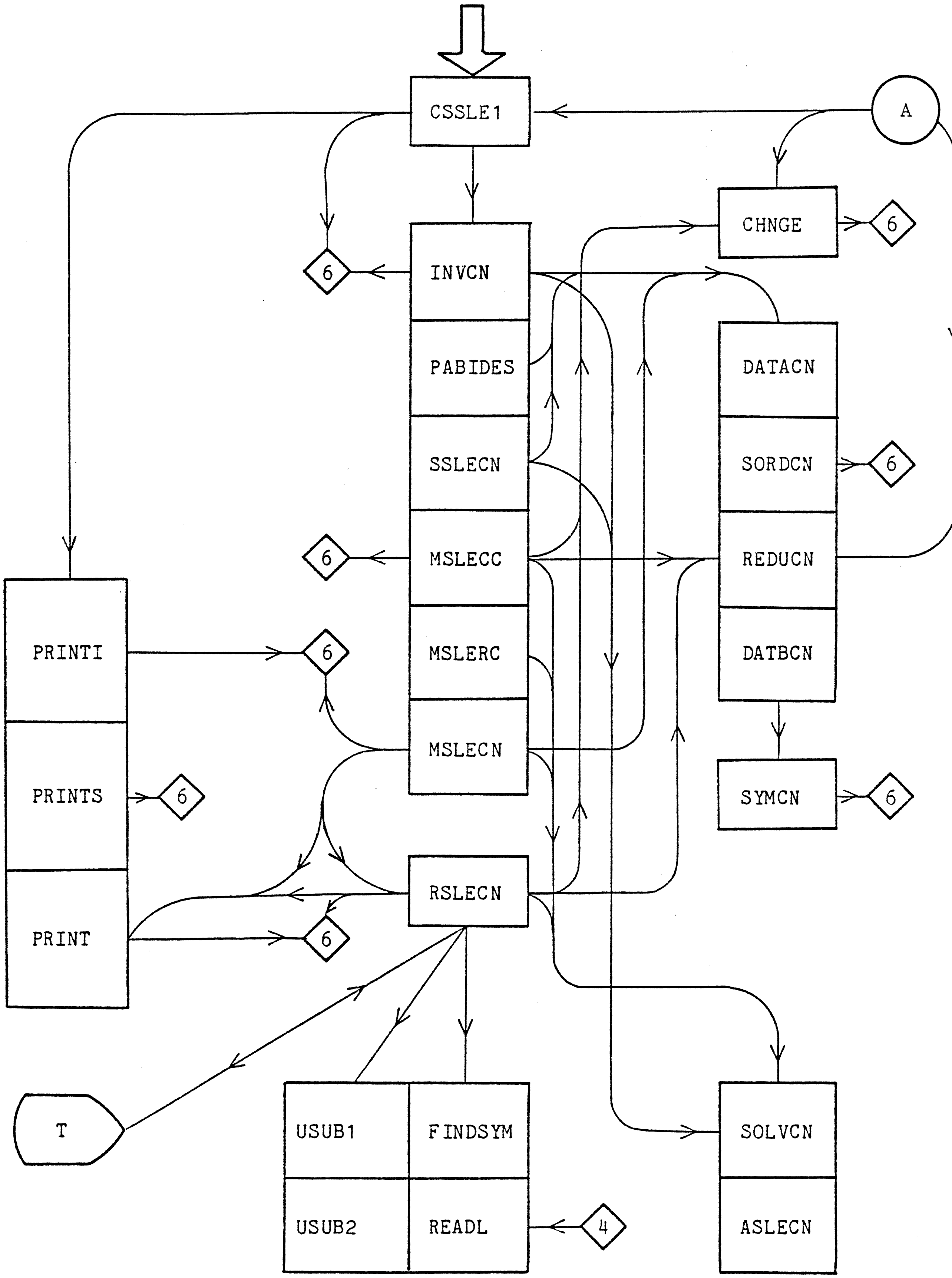


Fig. 9(a) Block diagram of the package.

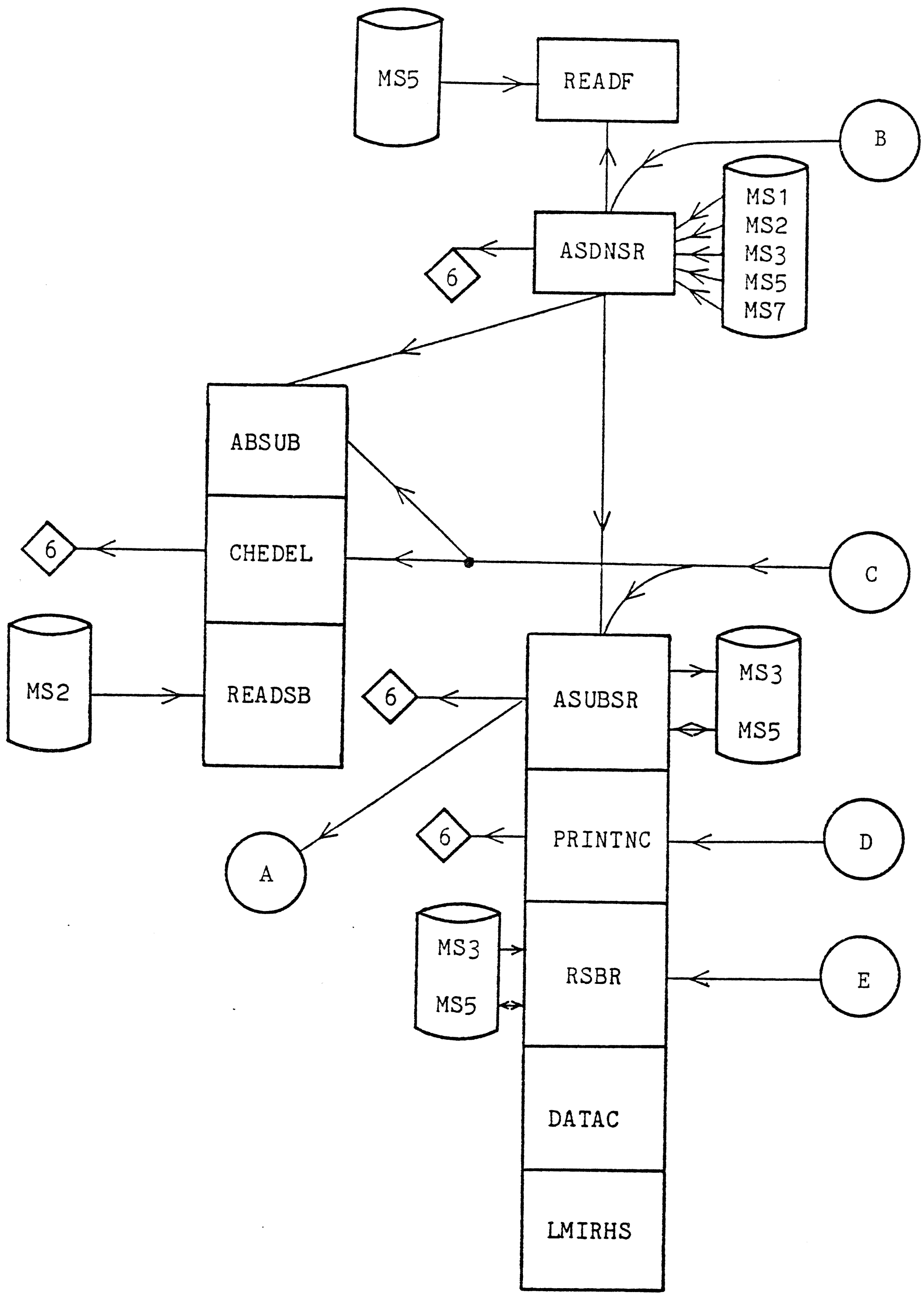

MS5
READF
B
MS1
MS2
MS3
MS5
MS7
ASDNSR
6
ABSUB
CHEDEL
6
C
MS2
READSB
6
ASUBSR
MS3
MS5
A
6
PRINTNC
D
MS3
MS5
RSBR
E
DATAC
LMIRHS

Fig. 9(b)

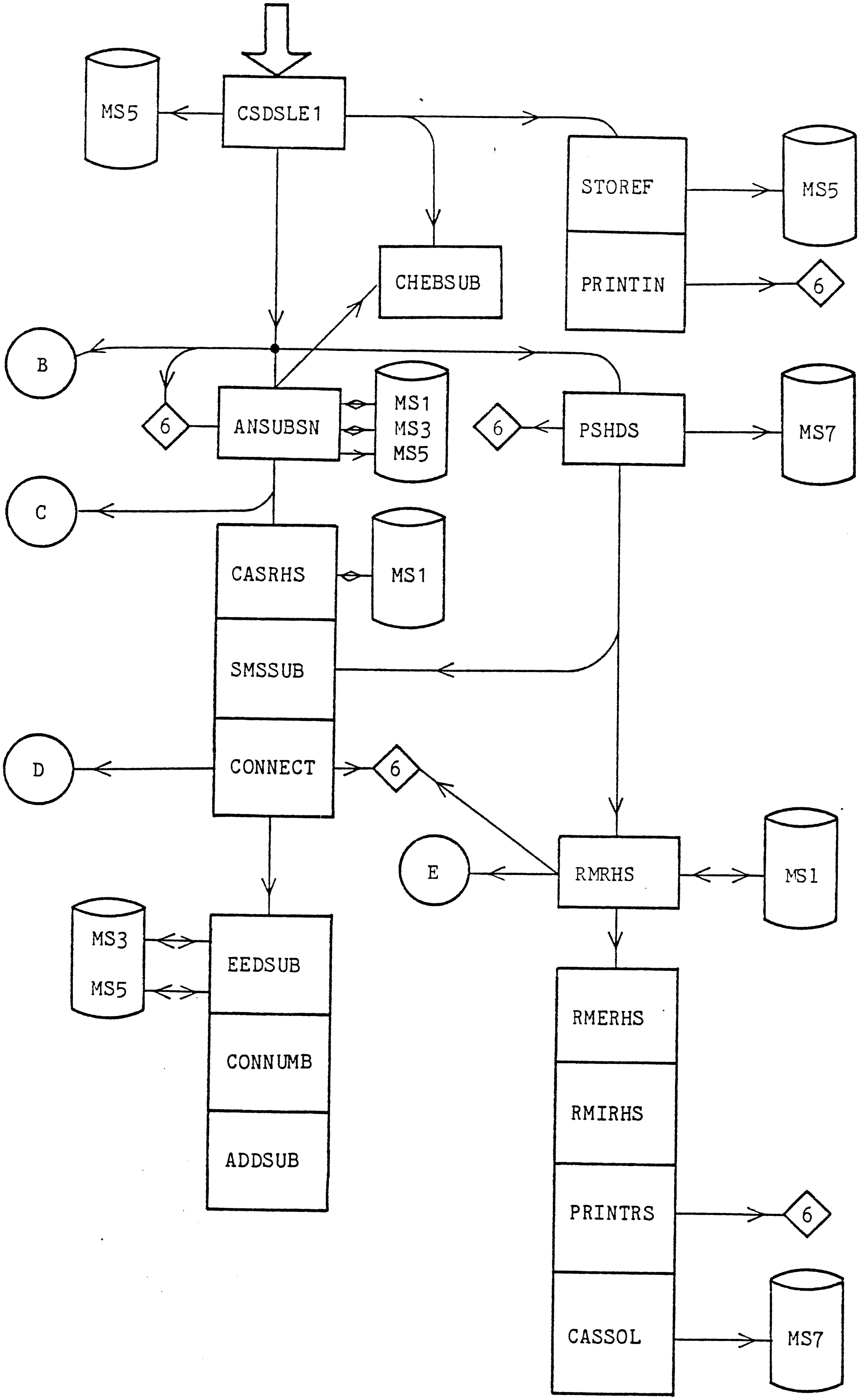

MS5
CSDSLE1
STOREF
MS5
CHEBSUB
PRINTIN
6
B
6
ANSUBSN
MS1
MS3
MS5
6
PSHDS
MS7
C
CASRHS
MS1
SMSSUB
D
CONNECT
6
E
RMRHS
MS1
MS3
MS5
EEDSUB
CONNUMB
ADDSUB
RMERHS
RMIRHS
PRINTRS
6
CASSOL
MS7

Fig. 9(c)

is printed out and execution of the main subroutine is stopped with IN(3)=1. If the user updates some coefficients in the submatrix with row and column indices outside the nonzero pattern then the message

ELEMENT OUT OF THE MATRIX AREA

is printed out and execution of the main subroutine is stopped with IN(3)=1.

Partial bifactorization of decomposed submatrices for internal nodes of a subgraph is performed in PABIDES called from CSSLE1 with IN(8)=6. For a description of other options with IN(8)=1, ..., 5, see [3]. DATAC, DATACN and DATBCN are used to prepare the proper form of data for library subroutines. SYMCN checks the structural symmetry and ordering of columns of the coefficient matrix. If the matrix is not symmetrical, new zero coefficients are added to create symmetry. The user is informed by the statement

INITIAL COEFFICIENT MATRIX NONSYMMETRICAL

Each column is checked to see whether the elements 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

SORDCN and REDUCN simulate, order and execute Gauss elimination as described in [2]. REDUCN is also used to perform partial bifactorization for the submatrix, corresponding to internal nodes of its subgraph only.

SOLVCN and ASLECN solve the linear equations by a sequence of matrix multiplications (2) for the original and adjoint (transpose) system respectively.

CHNGE changes the value of one element in the coefficient matrix.

If the user tries to update a coefficient which was not previously stored, he will be notified by the statement

ELEMENT FROM ROW rnr AND COLUMN cnr WAS NOT STORED IN THE PREVIOUS COEFFICIENT MATRIX

where rnr, cnr denote row and column indices, respectively.

Subroutines PRINTIN, PRINTNC, PRINTRS, PRINT, PRINTI and PRINTS are used to print output information on the file No. 6.

LMIRHS multiplies the left part of the LR factorization by the right-hand side (RHS) vector $\underset{\sim}{x}$ according to the formula

$$\underset{\sim}{z}_j = \xi_{I_k I_j} \circ \xi_{I_{k-1} I_j} \circ \ldots \circ \xi_{I_{j+1} I_j} \circ \xi_{I_j I_j} \circ \underset{\sim}{x} , \qquad (26)$$

where j is the index of the subgraph describing a given submatrix and $\xi_{I_i I_j}$ is defined in (12). Subroutine ANSUBSN organizes the partial analysis process for updated submatrices corresponding to the partition nodes of middle graphs, according to the notation described in Section II. These submatrices, called interconnection matrices, have their corresponding graphs called substitute graphs. Only those substitute graphs are analyzed which are necessary for updating the results of a previous solution if the system was previously solved.

A sequence of modified substitute graphs to be reanalyzed is determined in SMSSUB. CONNECT combines the results of analysis of two subgraphs to obtain the description of substitute graph. EEDSUB is used to extract the external description of subgraphs represented at the partition nodes of their ascendant. CONNUMB generates original indices of the nodes for substitute graph. ADDSUB adds two submatrices to obtain the interconnection matrix. CASRHS calculates addresses and stores the RHS vector for the substitute graph. CASSOL calculates addresses and stores the solution vector for a subgraph.

The partial solution at the nodes incident with updated submatrices is calculated in PSHDS. RMRHS organizes multiplication of the right part of the LR factorization by the $\underset{\sim}{z}$ vector (15) according to the formula

$$\underset{\sim}{y}_j = \Phi_{I_j I_j} \circ \Phi_{I_j I_{j+1}} \circ \dots \circ \Phi_{I_j I_k} \circ \underset{\sim}{z} , \qquad (27)$$

where j is the index of the subgraph describing a set of nodes where the solution is calculated $\Phi_{I_j I_i}$ is defined in (13). RMERHS executes a part of (27) for $\Phi_{I_j I_i}$, $i \in \{j+1,\dots,k\}$ and RMIRHS executes the last of (27), i.e., the multiple product with $\Phi_{I_j I_j}$.

IV. LIST OF ARGUMENTS

Standard entry (subroutine CSDSLE1)

The subroutine call is

```
CALL CSDSLE1(IN,INTWA,CMPLXWA,IM,SOLR)
```

The arguments are as follows:

IN is an integer indicator vector. Its length is at least

IN(4) + IN(5) + IN(9)/2 + 26.

Arguments of IN have the following values:

IN(1) =N, number of unknowns in the current subgraph.

IN(2) =LES, maximum area predicted for the sparse matrix describing the current subgraph.

IN(3) =IAR, flag for insufficient area.

IN(4) number of subgraphs where the solution will be recalculated.

IN(5) the number of all proper blocks.

IN(6) =NINT, the number of internal nodes in the current subgraph.

IN(7) indicator for printing intermediate results

=0 only part of the intermediate results will be printed,

=1 results will not be printed,

=2 all intermediate results will be printed,

=3 only the solution will be printed.

IN(8) the indicator for the kind of job

=6 the complete bifactorization process is executed for all proper blocks,

=7 the solution is calculated for an altered coefficient matrix and/or right-hand side vector (previous use of the main subroutine CSDSLE1 with IN(8)=6 is required),

=8 the solution is calculated for an altered right-hand side vector (previous use of the main subroutine CSDSLE1 with IN(8)=6 is required).

IN(9) =MNS, the highest index of the subgraphs (according to the decomposition tree.

IN(10) =NMS, the number of modified subgraphs, must be less than IN(12)/2.

IN(11) =LREC, the length of the basic record in double CM words, must be greater than 256.

IN(12) =LADR, the length of the addressing record in CM words, must be greater than 2*NMS.

IN(13) current number of stored basic records.

IN(14) =NSBR, current number of subgraphs in the basic record.

IN(15) maximum predicted number of basic and solution records to store the results of analysis of all subgraphs.

IN(16) maximum number of equations for any subgraph.

IN(17) maximum area predicted for sparse matrix analysis of subgraphs, usually 1-2 times the number of nonzero elements in the maximum submatrix.

IN(18) the number of nonzero elements in the maximum submatrix.

IN(19) the lowest index over all decomposition nodes.

IN(20) the number of data records storing the information about modified subgraphs.

IN(21) current number of RHS records.

IN(22) the indicator for the RHS updating formula

=0 the complete solution will be recalculated,

=1 the solution for modified proper blocks only will be recalculated,

=2 only the solution for the subgraphs specified by the user will be recalculated - if the user specifies substitute subgraphs only then the solution will be recalculated at the partition nodes only.

IN(23) the indicator for operations on random files

=0 random access multi-records files will be created,

=1 random access multi-records files will be updated,

=2 the program will be executed without creating random access multi-records files,

=3 random access multi-records files will be extended.

IN(24) current number of stored addressing records.

IN(25) the length of data and RHS records in double CM words, must greater than 2*IN(10).

IN(26) number of all decomposition modes.

IN(27)
-IN(L) store numbers of external nodes for the substitute subgraphs

where L = 26 + IN(9)/2.

If the substitute subgraph is spanned over the partition nodes of the graph number NG, then the number of its external nodes (partition nodes of the ascendant) must be stored in IN(26+NG). Obviously, substitute graph number 1 will always have a number of external nodes equal 0, so IN(27)=0.

IN(L+1) -IN(M) store the indices of all proper blocks in decreasing order, where M=L+IN(5).

IN(M+1) -IN(K) store the indices of subgraphs, where the solution will be calculated, in decreasing order, where K=M+IN(4).

IM(1) =MS1, mass storage indicator

=0 mass storage files will not be used,

=1 mass storage files will be used,

=2 mass storage file No. 2 (data file) will not be used,

=3 mass storage files will be extended.

IM(2) =NRI, flag denoting a reference to the main subroutine, must be initialzed by the user to 0 in conjunction with the first call and must never be subsequently altered.

IM(3) the size of working area INTWA.

IM(4) the size of working area CMPLXWA.

IM(5) the size of the solution record in double CM words. At the output IM(5) contains the number of elements of the solution vector.

INTWA is an INTEGER working area. Its length is at least

9*IN(9)/2 + (IN(9)-1)/IN(12) + IN(9)/50 + 2*IN(12) + 2*IN(15) +

4*IN(16) + 2*(IN(17) + IN(18)) + IN(20) + (IN(26)/IN(25)/+1)*2 +

19.

For small problems of up to 10 submatrices, the user is advised to set standard values IN(9)=20, IN(12)=20, IN(15)=11, IN(20)=10, and use INTWA of length

4*IN(16) + 2*(IN(17) + IN(18)) + 181.

CMPLXWA is a COMPLEX working area. Its length is at least

1 + IN(11) + 3*IN(16) + 4*IN(17) + IN(18) + IN(25).

SOLR solution record. For small problems, it contains the solution vector. In this case, its size must be not less than the total number of variables.

Entry for analysis without decomposition (subroutine CSSLE1)

For a description of subroutine parameters and different job options, see [3].

V. DATA STRUCTURE

There are two different forms of data describing the decomposed system. The first is used principally when initial information about large decomposed systems is needed to obtain the first or nominal solution. In this form, data is assumed to be stored in random access multi-record files in the local file No. 2. Each record contains IN(25) complex numbers. The second is used when the system is not so large or principally when the changes in the coefficient matrix or the right-hand side vector are made in a local area and information about those changes can be stored in one data record. This data record can be stored directly in the working area CMPLXWA from element [IN(11) + 3*IN(16) + 4*IN(17) + 1]. This will save time needed to execute the mass storage read subroutine. The structure of data file No. 2 is shown in Fig. 10.

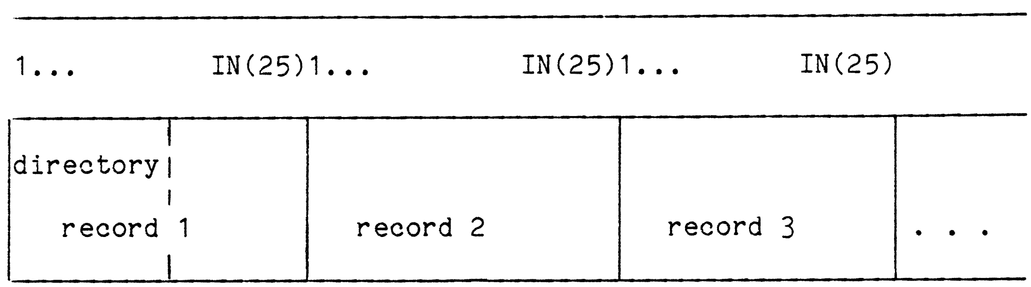


Fig. 10. Structure of data file.

The first data record DREC in file No. 2 contains the <u>directory</u> for all the modified subgraphs. The directory is a COMPLEX vector of length 2*IN(10). Information about the NBSth modified subgraph is described by four integer numbers A, B, C, D, stored in the real and imaginary parts of the DREC elements as follows:

DREC(NBS) = CMPLX(A,B)

DREC(NBS + IN(10)) = CMPLX(C,D)

where

A is the original index of the subgraph,

B is the index of the record containing information about modifications in the subgraph A,

C is the length of complex area storing this information,

D is the address in the record B where this information is stored.

The information about modifications in the subgraphs is stored in the form of COMPLEX matrices CE starting from element 2*IN(10)+1 in record 1 and starting from element 1 in other records.

There are two different forms of information, depending on whether the changes are done in the coefficient matrix or the right-hand side vector.

Data structure for altered coefficient matrix

This structure is also used when the first or nominal solution is calculated. A COMPLEX matrix CE describing the modified subgraph has the structure shown in Fig. 11.

1	2	NEL	NEL	N	(N-1)/2 + 1
N	LES	AK	NROW	V	NON(1), NON(3), NON(5), ...
NINT	NEL		NCOL		NON(2), NON(4), NON(6), ...

Fig. 11. Structure of a matrix CE for updated coefficient matrix.

The first two complex elements of CE are as follows:

CE(1) = CMPLX(N,NINT)

CE(2) = CMPLX(LES,NEL)

where

N is the number of nodes in the subgraph describing the altered matrix,

NINT is the number of internal nodes (those which are not partition nodes) in the subgraph,

LES is the length of the area predicted for the sparse matrix operations (2 to 5 times more than NEL),

NEL is the number of updated elements in the submatrix (for the first analysis of a subnetwork NEL is equal to the number of nonzero elements in the submatrix).

Next NEL elements of CE contain the nonzero coefficients of the submatrix. NEL elements starting from the elements NEL+3 contain row and column indices of consecutive nonzero coefficients, stored in the real and imaginary parts of the CE elements, respectively. Row and column indices must refer to the internal numbering of nodes within each subgraph, i.e., they must be numbered consecutively starting from 1 up to N. A COMPLEX vector V stored in CE starting from the element 2*NEL+3 contains the right-hand side (RHS) vector, and finally, the INTEGER vector NON stored in CE starting from the element 2*NEL+N+3 contains the original indices of the nodes as obtained after graph partition. Elements of NON are stored consecutively in the real and imaginary parts of the CE elements, as illustrated in Fig. 11. While updating the coefficient matrix and RHS vector (with IN(8) = 7), the user may specify as many changes in the coefficient matrix as he wants to by storing NEL updated elements. If a certain element is updated in more than one place, the latest value will replace the old element in the submatrix.

In this case, (IN(8) = 7) the user may not store the vector NON, as these values should not be changed during successive solutions. New values of the whole vector V must be stored even if its value is not updated.

<u>Data structure for altered RHS only</u>

Previous use of the main subroutine with IN(8) = 6 is required before this kind of data can be accepted by the program. When the main subroutine is run with IN(8) = 8, data records are organized as shown in Fig. 10 with the directory in record 1. Each subgraph remains unchanged and only the RHS vector is updated. Complex matrix CE describing the modified subgraph only contains the updated RHS vector as shown in Fig.

12. Again, the entire vectors V must be stored for all those subgraphs for which RHS is updated.

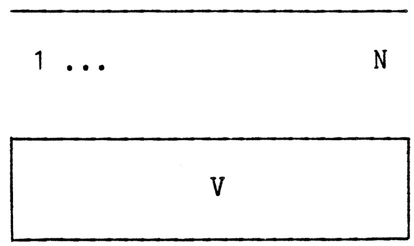


Fig. 12. Structure of a matrix CE for updated RHS only.

Data describing the graph decomposition and the type of job

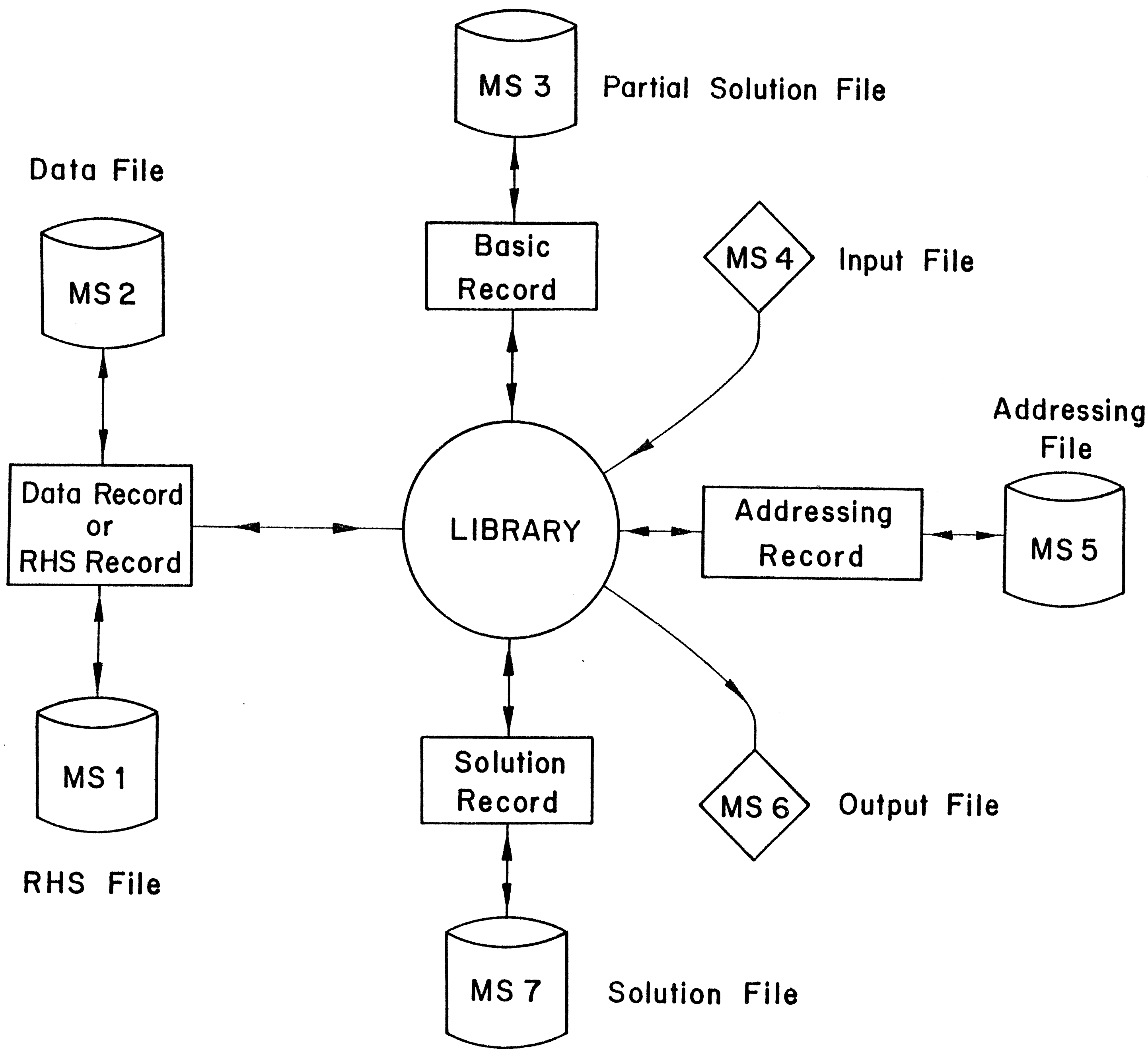
In the vector IN, the user must assign values to the following variables: IN(4), IN(5), IN(7), IN(8), IN(9), IN(10), IN(11), IN(12), IN(15), IN(16), IN(17), IN(18), IN(19), IN(20), IN(22), IN(23), IN(25), up to IN(K), where

$$K = IN(4) + IN(5) + IN(9)/2 + 26.$$

The user must decide whether or not he will use or create random access multi-record files. These files are necessary when the system is big, so that only part of the information describing it can be stored in central memory. Fig. 13 shows the scheme of memory organization as used by the library subroutines when random files are necessary.

Data records have been described at the beginning of this Section. Each basic record may store information about a partial solution of up to 255 subgraphs. Each subgraph in a basic record has specified its number of nodes N, the number of internal nodes NINT, the length of area predicted for the sparse matrix operations LES, a full description of its bifactorization as generated by the Zollenkopf algorithm [2], the

Fig. 13 Memory organization.



vector NON of the original indices of nodes, and the result Z of the multiple product (26) for the subnetwork. The structure of the information describing each subgraph as stored in a basic record is shown in Fig. 14.

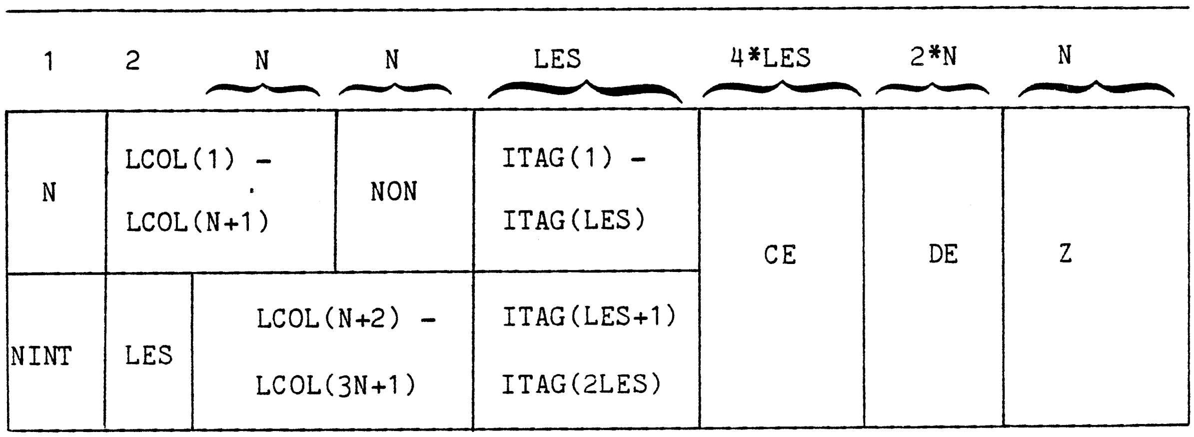


Fig. 14. Structure of the information describing a subgraph in a basic record.

Vectors LCOL, ITAG, CE and DE are as described in [3]. Each basic record has the structure shown in Fig. 15.

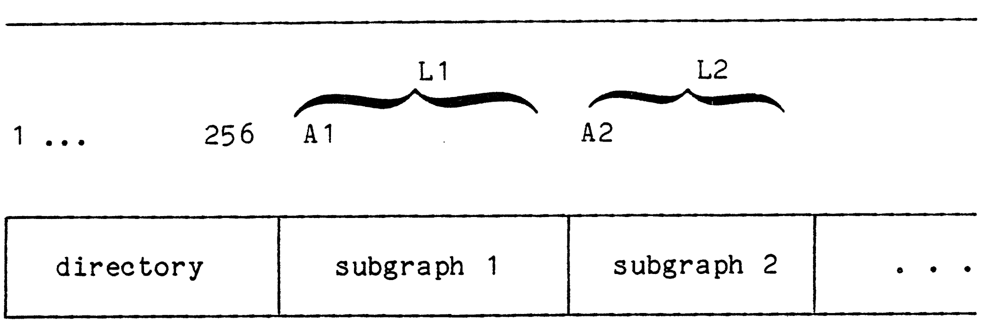


Fig. 15. Structure of a basic record.

The directory in a basic record is a COMPLEX vector of the length 256. Information about the kth subgraph (k = 1, ..., 255) stored in this basic record is described by two integer numbers Ak and Lk stored in the real and imaginary parts of the kth complex element of a basic record, where Ak denotes the address from where the information describing the kth subgraph starts and Lk is the length of the area storing the kth subgraph. Subgraphs are numbered consecutively as they are stored in a basic record, so k is not the original index of a subgraph according to the decomposition tree.

Original indices of graphs refer to the addressing records. Each addressing record is an INTEGER vector of length LDDR $\triangleq$ 2*LADR = 2*IN(12). If a graph's original index is NSUB, then its address can be found in the addressing record number NR = (NSUB-1)/LADR + 1. Element IADR = NSUB -(NR-1)*LADR of this record contains the index of the basic record storing the information about the subgraph NSUB, while element (IADR+LADR) is the index of the subnetwork in this basic record.

Addressing records are stored in random access file No. 5. The last record in file No. 5 contains information about stored multi-record random access files.

Multirecord random access file No. 1 contains information about the RHS vector at the nodes of substitute subgraphs (external nodes of proper blocks) or the solution vector at these nodes after the analysis is completed. Each RHS record is a COMPLEX vector whose length is IN(25). The first element in the first RHS record corresponds to the RHS at node IN(19). The number of RHS records is equal to IN(26)/IN(25) + 1. The next IN(26)/IN(25) + 1 records in file No. 1 store the $\underset{\sim}{z}$ vector (15) for the nodes of the substitute subgraphs and are used when CSDSLE1 is executed with IN(8) = 7 or IN(8) = 8.

The complete solution vector is stored in the multirecord random access file No. 7. The number of components of the calculated solution vector is available in IM(5). Each solution record contains pairs of complex numbers (A,B). The real part of each complex number A is the original index of a variable and B is the value of the solution for this variable.

VI. GENERAL INFORMATION

Use of COMMON: None.

Workspace: Provided by the user.

Input/output: Input from workspace or random access files. Output as defined by the user; see IN(7).

Subroutines: CSSLE1, DATACN, DATBCN, SYMCN, SORDCN, REDUCN, CHNGE, PRINT and:

a) for standard entry: CSDLSE1, ASDNSR, ABSUB, CHEBSUB, STOREF, READF, READSB, CHEDEL, ASUBSR, LMIRHS, ANSUBSN, SMSSUB, CONNECT, EEDSUB, CONNUMB, ADDSUB, CASRHS, PSHDS, RMRHS, RSBR, RMERHS, RMIRHS, CASSOL, PABIDES, DATAC, PRINTIN, PRINTNC, PRINTRS;

b) for entry for analysis without decomposition: SSLECN, MSLECN, INVCN, MSLECC, MSLERC, SOLVCN, ASLECN, RSLECN, FINDSYM, READL, PRINTI, PRINTS.

Restrictions: $IN(9) \geq IN(5) \geq IN(4) \geq 0$

$IN(5) \geq IN(10) \geq 0$, $IN(11) > 20$, $IN(12) > 2*IN(10)$

$IN(15) > 0$, $IN(16) > 0$, $IN(17) > 0$, $IN(18) > 0$,

$IN(19) > 0$, $IN(20) > 0$, $IN(25) > IN(9)$, $IN(26) \geq 0$

Date: January 1983.

VII. EXAMPLES

Example 6

the system of linear equations of Example 2 has been partitioned into 4 subsystems according to the partition of flow-graph G (see Fig. 6). Proper blocks G_4, G_5, G_6 and G_7 are represented as a data record in random access file No. 2. This file has been created by execution of the program DECSYS1 listed on the p. 48. As required by the Record Manager, the size of vector INDEX2 and the value of IREC2 must be not less than the number of data records +1. After file No. 2 (TAPE2) was created the main program EXAMP6 was executed. In this case the sizes of the basic record, the addressing record, and the data record were chosen sufficiently large and only one record of each type was created. All intermediate results are printed and the solution is calculated for all subgraphs. The user's program EXAMP6 and the results are shown on pp. 49-59. For each subgraph, the matrix $\underset{\sim}{F}$ of (7) is presented in the form of 3 matrices:

(1) Lower triangular part - elements of this matrix are equal to $-\bar{\ell}^j_{ij}/\bar{\ell}^j_{jj}$, i = (j + 1), ..., n and have opposite signs to those of $\underset{\sim}{L}^j$ of (6).

(2) Upper triangular part - elements of this matrix are equal to $-r^j_{ji}$, i = (j+1), ..., n and have opposite signs to those of $\underset{\sim}{R}^j$ of (4).

(3) Diagonal elements equal to d^j_{jj} as elements of $\underset{\sim}{D}^j$ of (6).

Compare the matrix $\underset{\sim}{F}$ obtained in Example 3 with the one described by the output shown on pp. 51-58.

```
      PROGRAM DECSYS1(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(2)
      COMPLEX CE(103)
C
C         DIRECTORY
C
      DATA CE/(4.,1.),(5.,1.),(6.,1.),(7.,1.),(13.,9.),(25.,22.),
     +(32.,47.),(25.,79.),
C
C         SUBGRAPH G4
C
     +(2.,1.),(2.,4.),(1.,0.),(2.,0.),(2.,0.),(2.,0.),
     +(1.,1.),(2.,1.),(1.,2.),(2.,2.),(1.,0.),(1.,0.),(6.,8.),
C
C         SUBGRAPH G5
C
     +(3.,1.),(6.,9.),(3.,0.),(1.,0.),(2.,0.),(2.,0.),(0.,0.),
     +(0.,0.),(1.,0.),(1.,0.),(4.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(0.,0.),(0.,0.),(9.,0.),
     +(5.,8.),(9.,0.),
C
C         SUBGRAPH G6
C
     +(4.,2.),(12.,12.),(6.,0.),(-1.,0.),(1.,0.),(3.,0.),(5.,0.)
     +,(-1.,0.),(2.,0.),(2.,0.),(0.,0.),(0.,0.),(1.,0.),(0.,0.),(1.,1.),
     +(2.,1.),(3.,1.),(1.,2.),(2.,2.),(3.,2.),(4.,2.),(1.,3.),(2.,3.),
     +(3.,3.),(2.,4.),(4.,4.),(0.,0.),(1.,0.),(0.,0.),(0.,0.),(3.,4.),
     +(7.,9.),
C
C         SUBGRAPH G7
C
     +(3.,2.),(6.,9.),(9.,0.),(1.,0.),(0.,0.),(2.,0.),(8.,0.),
     +(2.,0.),(1.,0.),(1.,0.),(7.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(-1.,0.),(2.,0.),(0.,0.),
     +(1.,2.),(7.,0.)/
C
C         IREC2    THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                  NOT LESS THAN IN(20)+1
C
      IREC2=2
      CALL OPENMS (2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS (2,CE,206,1)
      STOP
      END
```

```
      PROGRAM EXAMP6( INPUT,OUTPUT,RESULT,TAPE4= INPUT,TAPE6=RESULT,TAPE1,
     1TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 6
C
      DIMENSION IN(33), IM(5), INTWA( 137)
      COMPLEX CMPLXWA(716),SOLR(20)
      DATA IN/0,0,0,4,4,0,2,6,7,4,540,10,0,0,1,4,12,12,7,1,0,0,0,0,103,3
     1,0,1,1,7,6,5,4/
      DATA IM/1,0,137,716,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 6",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 6

HIERARCHICAL ANALYSIS OF LINEAR DECOMPOSED SYSTEMS (CSDSLE PACKAGE)

INPUT DATA

IN(4) - NUMBER OF SUBGRAPHS WHERE THE SOLUTION WILL BE RECALCULATED . .	4
IN(5) - NUMBER OF PROPER BLOCKS	4
IN(7) - INDICATOR FOR PRINTING INTERMEDIATE RESULTS	2
IN(9) - THE HIGHEST INDEX OF ALL SUBGRAPHS	7
IN(10) - NUMBER OF MODIFIED SUBGRAPHS	4
IN(11) - LENGTH OF BASIC RECORD	540
IN(12) - LENGTH OF ADDRESSING RECORD	10
IN(15) - MAXIMUM NUMBER OF BASIC RECORDS	1
IN(16) - MAXIMUM NUMBER OF EQUATIONS FOR THE SUBGRAPH	4
IN(17) - MAXIMUM AREA FOR SPARSE MATRIX CALCULATIONS	12
IN(18) - NUMBER OF NONZERO ELEMENTS IN THE MAXIMUM SUBMATRIX	12
IN(19) - THE LOWEST INDEX OF THE DECOMPOSITION NODES	7
IN(20) - NUMBER OF DATA RECORDS	1
IN(22) - INDICATOR FOR RHS UPDATING FORMULA	0
IN(23) - INDICATOR FOR OPERATIONS ON RANDOM FILES	0
IN(25) - MAXIMUM LENGTH OF DATA AND RHS RECORD	103
IN(26) - NUMBER OF ALL DECOMPOSITION NODES	3

NUMBERS OF EXTERNAL NODES IN THE SUBSTITUTE SUBGRAPHS

```
  0    1    1

INDICES OF ALL PROPER BLOCKS

  7    6    5    4

INDICES OF MODIFIED SUBGRAPHS

  4    5    6    7

COEFFICIENT MATRIX OF SUBGRAPH NO  4

  ROW  COL            VALUE                 ROW  COL            VALUE
                REAL          IMAGINARY                   REAL          IMAGINARY

   1    1    .100000E+01   0.                1    2    .200000E+01   0.
   2    1    .200000E+01   0.                2    2    .200000E+01   0.

CPU TIME:    .001 SECONDS

LOWER TRIANGULAR PART

  COL  ROW            VALUE                 ROW           VALUE
                REAL          IMAGINARY              REAL          IMAGINARY

   1
        2   .2000000E+01   0.

UPPER TRIANGULAR PART

  COL  ROW            VALUE                 ROW           VALUE
                REAL          IMAGINARY              REAL          IMAGINARY

   1
        2   .2000000E+01   0.

DIACONAL ELEMENTS

 VARIABLE             VALUE              VARIABLE            VALUE
                REAL          IMAGINARY                REAL          IMAGINARY

    1       .100000E+01   0.                2      -.200000E+01   0.
```

COEFFICIENT MATRIX OF SUBGRAPH NO 5

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.300000E+01	0.	1	2	.200000E+01	0.	1	3	.100000E+01	0.
2	1	.100000E+01	0.	2	2	0.	0.	2	3	.100000E+01	0.
3	1	.200000E+01	0.	3	2	0.	0.	3	3	.400000E+01	0.

CPU TIME: .003 SECONDS

LOWER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	.1000000E+01	0.	3	.2000000E+01	0.			
2	3	-.1333333E+01	0.						

UPPER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	.6666667E+00	0.	3	.3333333E+00	0.			
2	3	.6666667E+00	0.						

DIAGONAL ELEMENTS

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	.333333E+00	0.	2	-.666667E+00	0.	3	.333333E+01	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 6

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.600000E+01	0.	2	2	.500000E+01	0.	2	3	0.	0.
2	1	-.100000E+01	0.	3	2	-.100000E+01	0.	3	3	0.	0.
3	1	.100000E+01	0.	4	2	.200000E+01	0.	2	4	.100000E+01	0.
1	2	.300000E+01	0.	1	3	.200000E+01	0.	4	4	0.	0.

CPU TIME: .003 SECONDS

LOWER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	-.1000000E+01	0.	3	.1000000E+01	0.			
2	3	-.1500000E+01	0.	4	.2000000E+01	0.			
3	4	-.1212121E+00	0.						

UPPER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	.5000000E+00	0.	3	.3333333E+00	0.			
2	3	.6060606E-01	0.	4	.1818182E+00	0.			
3	4	.2727273E+00	0.						

DIAGONAL ELEMENTS

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	.166667E+00	0.	3	-.242424E+00	0.
2	.181818E+00	0.	4	-.363636E+00	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 7

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.900000E+01	0.	1	2	.200000E+01	0.	1	3	.100000E+01	0.
2	1	.100000E+01	0.	2	2	.800000E+01	0.	2	3	.100000E+01	0.
3	1	0.	0.	3	2	.200000E+01	0.	3	3	.700000E+01	0.

CPU TIME: .002 SECONDS

LOWER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	.1000000E+01	0.						
2	3	.2000000E+01	0.	3	0.	0.			

UPPER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1	2	.2222222E+00	0.						
2	3	.1142857E+00	0.	3	.1111111E+00	0.			

DIAGONAL ELEMENTS

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	.111111E+00	0.	2	.128571E+00	0.	3	.677143E+01	0.

SEQUENCE OF SUBSTITUTE SUBGRAPHS TO BE REANALYZED

3 2 1

EXTERNAL DESCRIPTION OF THE SUBGRAPH NO 6
NUMBER OF NODES: 2 NUMBER OF ELEMENTS: 4

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	-.242424E+00	0.	2	1	-.121212E+00	0.
2	2	-.363636E+00	0.	1	2	.272727E+00	0.

EXTERNAL DESCRIPTION OF THE SUBGRAPH NO 7
NUMBER OF NODES: 1 NUMBER OF ELEMENTS: 1

ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.677143E+01	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 3

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.652900E+01	0.	2	1	-.121212E+00	0.
2	2	-.363636E+00	0.	1	2	.272727E+00	0.

CPU TIME: .001 SECONDS

LOWER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1						
	2	-.1212121E+00	0.			

UPPER TRIANGULAR PART

COL	ROW	VALUE REAL	VALUE IMAGINARY	ROW	VALUE REAL	VALUE IMAGINARY
1						

```
      2    .4177165E-01   0.

DIAGONAL ELEMENTS

 VARIABLE              VALUE               VARIABLE             VALUE

                  REAL          IMAGINARY                  REAL          IMAGINARY

     1      .153163E+00    0.                   2      -.358573E+00    0.



EXTERNAL DESCRIPTION OF THE SUBGRAPH NO  4
NUMBER OF NODES:  1     NUMBER OF ELEMENTS:  1

  ROW  COL                VALUE

                   REAL          IMAGINARY

   1    1   -.200000E+01    0.

EXTERNAL DESCRIPTION OF THE SUBGRAPH NO  5
NUMBER OF NODES:  2     NUMBER OF ELEMENTS:  4

  ROW  COL              VALUE                ROW  COL              VALUE

                   REAL          IMAGINARY                    REAL          IMAGINARY

   1    1   -.666667E+00    0.                 2    1   -.133333E+01    0.
   2    2    .333333E+01    0.                 1    2    .666667E+00    0.

COEFFICIENT MATRIX OF SUBGRAPH NO  2

  ROW  COL              VALUE                ROW  COL              VALUE

                   REAL          IMAGINARY                    REAL          IMAGINARY

   1    1   -.266667E+01    0.                 2    1   -.133333E+01    0.
   2    2    .333333E+01    0.                 1    2    .666667E+00    0.

CPU TIME:     .002 SECONDS


LOWER TRIANGULAR PART

  COL  ROW              VALUE                ROW             VALUE

                   REAL          IMAGINARY              REAL          IMAGINARY
```

```
   1
        2  -.1333333E+01  0.
UPPER TRIANGULAR PART

  COL  ROW              VALUE                ROW              VALUE

                REAL            IMAGINARY               REAL            IMAGINARY

   1
        2  -.2500000E+00  0.
DIAGONAL ELEMENTS

 VARIABLE              VALUE             VARIABLE              VALUE

                REAL            IMAGINARY               REAL            IMAGINARY

    1     -.375000E+00   0.                  2      .300000E+01   0.



EXTERNAL DESCRIPTION OF THE SUBGRAPH NO  2
NUMBER OF NODES:  1     NUMBER OF ELEMENTS:  1

  ROW  COL               VALUE

                REAL            IMAGINARY

   1    1    .300000E+01   0.
EXTERNAL DESCRIPTION OF THE SUBGRAPH NO  3
NUMBER OF NODES:  1     NUMBER OF ELEMENTS:  1

  ROW  COL               VALUE

                REAL            IMAGINARY

   1    1   -.358573E+00   0.
COEFFICIENT MATRIX OF SUBGRAPH NO  1

  ROW  COL               VALUE

                REAL            IMAGINARY

   1    1    .264143E+01   0.
```

```
CPU TIME:     .001 SECONDS

DIAGONAL ELEMENTS
 VARIABLE                   VALUE
                 REAL            IMAGINARY
     1       .373583E+00    0.



SOLUTION WILL BE RECALCULATED AT THE INTERNAL NODES OF SUBGRAPHS
  7    6    5    4    3    2    1
SOLUTION
--------

IN SUBGRAPH NO    1
 VARIABLE                   VALUE
                 REAL            IMAGINARY
     9       .345698E+01    0.
IN SUBGRAPH NO    2
 VARIABLE                   VALUE
                 REAL            IMAGINARY
     8       .123924E+01    0.
IN SUBGRAPH NO    3
 VARIABLE                   VALUE
                 REAL            IMAGINARY
     7      -.185777E+00    0.
IN SUBGRAPH NO    4
```

```
VARIABLE             VALUE
               REAL          IMAGINARY
    6      -.147849E+01    0.
IN SUBGRAPH NO    5
VARIABLE             VALUE
               REAL          IMAGINARY
    5      -.197849E+01    0.
IN SUBGRAPH NO    6
VARIABLE             VALUE                VARIABLE             VALUE
               REAL          IMAGINARY                   REAL          IMAGINARY
    3       .279658E+00    0.                  4      -.435464E+00    0.
IN SUBGRAPH NO    7
VARIABLE             VALUE                VARIABLE             VALUE
               REAL          IMAGINARY                   REAL          IMAGINARY
    1      -.155505E+00    0.                  2       .292660E+00    0.
NUMBER OF VARIABLES IN THE SOLUTION VECTOR    9
CPU TIME:     .509 SECONDS
```

Example 7

The system of linear equations of Example 2 has been partitioned and stored in random access file No. 2, as for Example 6. The sizes of the basic record, the addressing record and the data record are the same as for Example 6. Only part of the intermediate results are printed out. The solution is calculated at the partition nodes only, which are internal nodes of the subgraphs 3, 2 and 1. The user's main program EXAMP7 and the corresponding results are shown on pp. 61-65.

```
      PROGRAM EXAMP7(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1,
     1TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 7
C
      DIMENSION IN(36), IM(5), INTWA(137)
      COMPLEX CMPLXWA(716),SOLR(20)
      DATA IN/0,0,0,3,4,0,0,6,7,4,540,10,0,0,1,4,12,12,7,1,0,2,0,0,103,3
     1,0,1,1,7,6,5,4,3,2,1/
      DATA IM/1,0,137,716,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 7",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 7

HIERARCHICAL ANALYSIS OF LINEAR DECOMPOSED SYSTEMS (CSDSLE PACKAGE)

INPUT DATA

IN(4)	- NUMBER OF SUBGRAPHS WHERE THE SOLUTION WILL BE RECALCULATED . .	3
IN(5)	- NUMBER OF PROPER BLOCKS	4
IN(7)	- INDICATOR FOR PRINTING INTERMEDIATE RESULTS	0
IN(9)	- THE HIGHEST INDEX OF ALL SUBGRAPHS	7
IN(10)	- NUMBER OF MODIFIED SUBGRAPHS	4
IN(11)	- LENGTH OF BASIC RECORD	540
IN(12)	- LENGTH OF ADDRESSING RECORD	10
IN(15)	- MAXIMUM NUMBER OF BASIC RECORDS	1
IN(16)	- MAXIMUM NUMBER OF EQUATIONS FOR THE SUBGRAPH	4
IN(17)	- MAXIMUM AREA FOR SPARSE MATRIX CALCULATIONS	12
IN(18)	- NUMBER OF NONZERO ELEMENTS IN THE MAXIMUM SUBMATRIX	12
IN(19)	- THE LOWEST INDEX OF THE DECOMPOSITION NODES	7
IN(20)	- NUMBER OF DATA RECORDS	1
IN(22)	- INDICATOR FOR RHS UPDATING FORMULA	2
IN(23)	- INDICATOR FOR OPERATIONS ON RANDOM FILES	0
IN(25)	- MAXIMUM LENGTH OF DATA AND RHS RECORD	103
IN(26)	- NUMBER OF ALL DECOMPOSITION NODES	3

NUMBERS OF EXTERNAL NODES IN THE SUBSTITUTE SUBGRAPHS

0 1 1

INDICES OF ALL PROPER BLOCKS

7 6 5 4

INDICES OF SUBGRAPHS WHERE SOLUTION WILL BE CALCULATED

3 2 1

INDICES OF MODIFIED SUBGRAPHS

4 5 6 7

COEFFICIENT MATRIX OF SUBGRAPH NO 4

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.100000E+01	0.	1	2	.200000E+01	0.
2	1	.200000E+01	0.	2	2	.200000E+01	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 5

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.300000E+01	0.	1	2	.200000E+01	0.	1	3	.100000E+01	0.
2	1	.100000E+01	0.	2	2	0.	0.	2	3	.100000E+01	0.
3	1	.200000E+01	0.	3	2	0.	0.	3	3	.400000E+01	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 6

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.600000E+01	0.	2	2	.500000E+01	0.	2	3	0.	0.
2	1	-.100000E+01	0.	3	2	-.100000E+01	0.	3	3	0.	0.
3	1	.100000E+01	0.	4	2	.200000E+01	0.	2	4	.100000E+01	0.
1	2	.300000E+01	0.	1	3	.200000E+01	0.	4	4	0.	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 7

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY

1	1	.900000E+01	0.	1	2	.200000E+01	0.	1	3	.100000E+01	0.
2	1	.100000E+01	0.	2	2	.800000E+01	0.	2	3	.100000E+01	0.
3	1	0.	0.	3	2	.200000E+01	0.	3	3	.700000E+01	0.

SEQUENCE OF SUBSTITUTE SUBGRAPHS TO BE REANALYZED

3 2 1

COEFFICIENT MATRIX OF SUBGRAPH NO 3

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.652900E+01	0.	2	1	-.121212E+00	0.
2	2	-.363636E+00	0.	1	2	.272727E+00	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 2

ROW	COL	VALUE REAL	VALUE IMAGINARY	ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	-.266667E+01	0.	2	1	-.133333E+01	0.
2	2	.333333E+01	0.	1	2	.666667E+00	0.

COEFFICIENT MATRIX OF SUBGRAPH NO 1

ROW	COL	VALUE REAL	VALUE IMAGINARY
1	1	.264143E+01	0.

SOLUTION WILL BE RECALCULATED AT THE INTERNAL NODES OF SUBGRAPHS

3 2 1

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

```
IN SUBGRAPH NO    2
 VARIABLE                  VALUE
                  REAL            IMAGINARY
     3       .123924E+01    0.
IN SUBGRAPH NO    3
 VARIABLE                  VALUE
                  REAL            IMAGINARY
     7      -.185777E+00    0.
NUMBER OF VARIABLES IN THE SOLUTION VECTOR    3
CPU TIME:     .230 SECONDS
```

Example 8

The system of linear equations of Example 2 has been partitioned and stored in random access file No. 2 as for Example 6. Smaller basic and addressing records were chosen and, in this example, 2 of each were necessary.

The data record was the same as for Example 6. Only the solution vector is printed out. The user's main program EXAMP8 and the corresponding results are shown on pp. 67-68.

```
      PROGRAM EXAMP8(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1,
     1TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 8
C
      DIMENSION IN(33), IM(5), INTWA(130)
      COMPLEX CMPLXWA(586),SOLR(20)
      DATA IN/0,0,0,4,4,0,3,6,7,4,410,5,0,0,2,4,12,12,7,1,0,0,0,0,103,3,
     10,1,1,7,6,5,4/
      DATA IM/1,0,130,586,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 8",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 8

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	VALUE IMAGINARY
8	.123924E+01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
7	-.185777E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	VALUE IMAGINARY
6	-.147849E+01	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE REAL	VALUE IMAGINARY
5	-.197849E+01	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
3	.279658E+00	0.	4	-.435464E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	-.155505E+00	0.	2	.292660E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 9

CPU TIME: .179 SECONDS

Example 9

The system of linear equations of Example 2 has been partitioned and stored in complex vector CE. The data record is read directly into the working area CMPLXWA, starting from element 4*IN(7) + 3*IN(16) + IN(11) +1. The main program EXAMP9 is executed without creating or updating any mass storage files. It is possible to read the data record directly into CMPLXWA and to create or update mass storage files. In the latter case, IM(1) must be equal to 2. Only the solution vector is printed out. The user's program EXAMP9 and the corresponding results are shown on pp. 70-71.

```
      PROGRAM EXAMP9(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1,
     +TAPE2,TAPE3,TAPE5,TAPE7)
C
C        THIS IS THE MAIN PROGRAM FOR EXAMPLE 9
C
      DIMENSION IN(33),IM(5),INTWA(134)
      COMPLEX CMPLXWA(716),SOLR(20),CE(103)
C
C        DIRECTORY
C
      DATA CE/(4.,1.),(5.,1.),(6.,1.),(7.,1.),(13.,9.),(25.,22.),
     +(32.,47.),(25.,79.),
C
C        SUBGRAPH G4
C
     +(2.,1.),(2.,4.),(1.,0.),(2.,0.),(2.,0.),(2.,0.),
     +(1.,1.),(2.,1.),(1.,2.),(2.,2.),(1.,0.),(1.,0.),(6.,8.),
C
C        SUBGRAPH G5
C
     +(3.,1.),(6.,9.),(3.,0.),(1.,0.),(2.,0.),(2.,0.),(0.,0.),
     +(0.,0.),(1.,0.),(1.,0.),(4.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(0.,0.),(0.,0.),(9.,0.),
     +(5.,3.),(9.,0.),
C
C        SUBGRAPH G6
C
     +(4.,2.),(12.,12.),(6.,0.),(-1.,0.),(1.,0.),(3.,0.),(5.,0.)
     +,(-1.,0.),(2.,0.),(2.,0.),(0.,0.),(0.,0.),(1.,0.),(0.,0.),(1.,1.),
     +(2.,1.),(3.,1.),(1.,2.),(2.,2.),(3.,2.),(4.,2.),(1.,3.),(2.,3.),
     +(3.,3.),(2.,4.),(4.,4.),(0.,0.),(1.,0.),(0.,0.),(0.,0.),(3.,4.),
     +(7.,9.),
C
C        SUBGRAPH G7
C
     +(3.,2.),(6.,9.),(9.,0.),(1.,0.),(0.,0.),(2.,0.),(3.,0.),
     +(2.,0.),(1.,0.),(1.,0.),(7.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(-1.,0.),(2.,0.),(0.,0.),
     +(1.,2.),(7.,0.)/
      DATA IN/0,0,0,4,4,0,3,6,7,4,540,10,0,0,0,4,12,12,7,0,0,0,2,0,103,3
     +,0,1,1,7,6,5,4/
      DATA IM/0,0,134,716,10/
C
      IA=4*IN(17)+3*IN(16)+IN(11)
      DO 10 I=1,103
   10 CMPLXWA(IA+I)=CE(I)
      WRITE(6,20)
   20 FORMAT(1H ,/," EXAMPLE 9",/)
      CALL CSDSLE1(IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 9

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	VALUE IMAGINARY
8	.123924E+01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
7	-.185777E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	VALUE IMAGINARY
6	-.147849E+01	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE REAL	VALUE IMAGINARY
5	-.197849E+01	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
3	.279658E+00	0.	4	-.435464E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	-.155505E+00	0.	2	.292660E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 9

CPU TIME: .083 SECONDS

Example 10

The system of linear equations of Example 2 has been partitioned into 4 subsystems according to the partition of flow-graph G (see Fig. 6). Proper blocks G_4, G_5 and the directory are stored in the first data record in random access file No. 2, while proper blocks G_6 and G_7 are stored in the second data record. File No. 2 has been created by execution of the program DECSYS2 listed on p. 73. After creating file No. 2 the main program EXAMP10 was executed. Smaller basic and addressing records were chosen. Only the solution vector is printed out. The user's program EXAMP10 and the corresponding results are shown on pp. 74–75.

```
      PROGRAM DECSYS2(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(3)
      COMPLEX CE(46),DE(57)
C
C         DIRECTORY
C
      DATA CE/(4.,1.),(5.,1.),(6.,2.),(7.,2.),(13.,9.),(25.,22.),
     +(32.,1.),(25.,33.),
C
C         SUBGRAPH G4
C
     +(2.,1.),(2.,4.),(1.,0.),(2.,0.),(2.,0.),(2.,0.),
     +(1.,1.),(2.,1.),(1.,2.),(2.,2.),(1.,0.),(1.,0.),(6.,8.),
C
C         SUBGRAPH G5
C
     +(3.,1.),(6.,9.),(3.,0.),(1.,0.),(2.,0.),(2.,0.),(0.,0.),
     +(0.,0.),(1.,0.),(1.,0.),(4.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(0.,0.),(0.,0.),(9.,0.),
     +(5.,8.),(9.,0.)/
C
C         SUBGRAPH G6
C
      DATA DE/(4.,2.),(12.,12.),(6.,0.),(-1.,0.),(1.,0.),(3.,0.),(5.,0.)
     +,(-1.,0.),(2.,0.),(2.,0.),(0.,0.),(0.,0.),(1.,0.),(0.,0.),(1.,1.),
     +(2.,1.),(3.,1.),(1.,2.),(2.,2.),(3.,2.),(4.,2.),(1.,3.),(2.,3.),
     +(3.,3.),(2.,4.),(4.,4.),(0.,0.),(1.,0.),(0.,0.),(0.,0.),(3.,4.),
     +(7.,9.),
C
C         SUBGRAPH G7
C
     +(3.,2.),(6.,9.),(9.,0.),(1.,0.),(0.,0.),(2.,0.),(8.,0.),
     +(2.,0.),(1.,0.),(1.,0.),(7.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(-1.,0.),(2.,0.),(0.,0.),
     +(1.,2.),(7.,0.)/
C
C         IREC2    THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                  NOT LESS THAN IN(20)+1
C
      IREC2=3
      CALL OPENMS (2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS (2,CE,92,1)
      CALL WRITMS (2,DE,114,2)
      STOP
      END
```

```
      PROGRAM EXAMP10(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 10
C
      DIMENSION IN(33), IM(5), INTWA(131)
      COMPLEX CMPLXWA(540),SOLR(20)
      DATA IN/0,0,0,4,4,0,3,6,7,4,410,5,0,0,2,4,12,12,7,2,0,0,0,0,57,3,0
     1,1,1,7,6,5,4/
      DATA IM/1,0,131,540,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 10",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 10

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	VALUE IMAGINARY
8	.123924E+01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
7	-.185777E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	VALUE IMAGINARY
6	-.147349E+01	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE REAL	VALUE IMAGINARY
5	-.197849E+01	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
3	.279658E+00	0.	4	-.435464E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	-.155505E+00	0.	2	.292660E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 9

CPU TIME: .192 SECONDS

Example 11

The system of linear equations of Example 2 has been partitioned into 4 subsystems according to the partition of flowgraph G_1 (see Fig. 6). Proper blocks G_4, G_5 and the directory describing these two blocks only, are stored in the first data record in random access file No. 2. File No. 2 has been created by execution of the program DECSYS3 listed on p. 77.

After creating file No. 2, the main program EXAMP11 was executed. In this program, IN(10)=2, which means that two subgraphs will be analyzed. As a result, local files called TAPE1, TAPE3 and TAPE5 have been generated by the program. These files store information about the partial analysis of the proper blocks G_4 and G_5 and they must be preserved by the user to complete analysis after the rest of the network is analyzed.

The remaining part of the system is represented by proper blocks G_6 and G_7 stored in file No. 2 after execution of the program DECSYS4 listed on the p. 78.

Next, the program EXAM11A was executed. This program uses information about the system stored in files 1, 2, 3 and 5. Indicators IN(5), IN(9), IN(12), IN(15) up to IN(20), IN(25) and IN(26) must not be changed in EXAM11A as they determine the partitioning of the working areas and indicators evaluated in the previous run. User's programs EXAMP11, EXAMP11A and the corresponding results are shown on pp. 79-82.

```
      PROGRAM DECSYS3(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(2)
      COMPLEX CE(42)
C
C         DIRECTORY
C
      DATA CE/(4.,1.),(5.,1.),(13.,5.),(25.,18.),
C
C         SUBGRAPH G4
C
     +(2.,1.),(2.,4.),(1.,0.),(2.,0.),(2.,0.),(2.,0.),
     +(1.,1.),(2.,1.),(1.,2.),(2.,2.),(1.,0.),(1.,0.),(6.,8.),
C
C         SUBGRAPH G5
C
     +(3.,1.),(6.,9.),(3.,0.),(1.,0.),(2.,0.),(2.,0.),(0.,0.),
     +(0.,0.),(1.,0.),(1.,0.),(4.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(0.,0.),(0.,0.),(9.,0.),
     +(5.,8.),(9.,0.)/
C
C         IREC2    THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                  NOT LESS THAN IN(20)+1
C
      IREC2=2
      CALL OPENMS(2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS(2,CE,100,1)
      STOP
      END
```

```
      PROGRAM DECSYS4(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(2)
      COMPLEX CE(61)
C
C         DIRECTORY
C
      DATA CE/(6.,1.),(7.,1.),(32.,5.),(25.,37.),
C
C         SUBGRAPH G6
C
     +(4.,2.),(12.,12.),(6.,0.),(-1.,0.),(1.,0.),(3.,0.),(5.,0.)
     +,(-1.,0.),(2.,0.),(2.,0.),(0.,0.),(0.,0.),(1.,0.),(0.,0.),(1.,1.),
     +(2.,1.),(3.,1.),(1.,2.),(2.,2.),(3.,2.),(4.,2.),(1.,3.),(2.,3.),
     +(3.,3.),(2.,4.),(4.,4.),(0.,0.),(1.,0.),(0.,0.),(0.,0.),(3.,4.),
     +(7.,9.),
C
C         SUBGRAPH G7
C
     +(3.,2.),(6.,9.),(9.,0.),(1.,0.),(0.,0.),(2.,0.),(8.,0.),
     +(2.,0.),(1.,0.),(1.,0.),(7.,0.),(1.,1.),(2.,1.),(3.,1.),(1.,2.),
     +(2.,2.),(3.,2.),(1.,3.),(2.,3.),(3.,3.),(-1.,0.),(2.,0.),(0.,0.),
     +(1.,2.),(7.,0.)/
C
C          IREC2     THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                    NOT LESS THAN IN(20)+1
C
      IREC2=2
      CALL OPENMS(2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS(2,CE,122,1)
      STOP
      END
```

```
      PROGRAM EXAMP11(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 11
C
      DIMENSION IN(33), IM(5), INTWA(137)
      COMPLEX CMPLXWA(674),SOLR(20)
      DATA IN/0,0,0,2,4,0,3,6,7,2,540,10,0,0,1,4,12,12,7,1,0,1,0,0,61,3,
     10,1,1,7,6,5,4/
      DATA IM/1,0,137,674,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 11",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 11

HIERARCHICAL STRUCTURE IS NOT COMPLETE
SOLUTION CANNOT BE CALCULATED

```
      PROGRAM EXAM11A(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     +,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 11A
C
      DIMENSION IN(33),IM(5),INTWA(137)
      COMPLEX CMPLXWA(674),SOLR(20)
      DATA IN/0,0,0,2,4,0,3,6,7,2,540,10,0,0,1,4,12,12,7,1,0,1,3,0,61,3
     +,0,1,1,7,6,5,4/
      DATA IM/3,0,137,674,10/
C
      WRITE(6,100)
  100 FORMAT(1H ,/," EXAMPLE 11A",/)
      CALL CSDSLE1(IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 11A

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
7	-.185777E+00	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
3	.279658E+00	0.	4	-.435464E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
1	-.155505E+00	0.	2	.292660E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 6

CPU TIME: .134 SECONDS

Example 12

To test networks of different size and different partition types, the program LADDER, listed on pp. 86-89, was used. LADDER generates data for a decomposed system of linear equations which are the nodal equations of a ladder network. As input data for LADDER, the user must supply 3 integer numbers LEVEL, NN and IN25, where LEVEL is the number of decomposition levels, NN is a number of nodes in each proper block, and IN25 is the length of the data record. As a result, LADDER will generate data, which is stored in file No. 2 in the form required by the subroutine CSDSLE1. To prepare data for the user's main program EXAMP12, LADDER was executed with input data as follows: LEVEL = 2, NN = 10, and IN25 = 1500. At the output, LADDER writes the following information:

LENGTH OF WORKING AREA INTWA MUST BE NOT LESS THAN IA7

LENGTH OF WORKING AREA CMPLXWA MUST BE NOT LESS THAN IB5

NUMBER OF DATA RECORDS + 1 = IREC2

NUMBER OF ALL NODES = NALL

NUMBER OF ALL PROPER BLOCKS = IN5

THE HIGHEST INDEX OF ALL THE SUBGRAPHS = IN9

MAXIMUM AREA FOR SPARSE MATRIX = IN17

NUMBER OF NONZERO ELEMENTS IN MAX SUBMATRIX = IN18

THE LOWEST INDEX OVER ALL DECOMPOSITION NODES = IN19

NUMBER OF ALL DECOMPOSITION NODES = IN26

This information is necessary to run the user's program EXAMP12, where indicator vector IN must be as follows:

IN(5) = IN5, IN(9) = IN9, IN(10) = IN5, IN(16) = NN, IN(17) = IN17, IN(18) = IN18, IN(19) = IN19, IN(20) = IREC2, IN(25) = IN25, IN(26) = IN26.

With LEVEL = 2, NN = 10, and IN25 = 1500, LADDER generated the decomposed graph of a ladder network having 37 nodes, partitioned into 4 proper blocks. Each proper block represents a section of the ladder network having 10 nodes (plus reference), 18 unit resistors and one unit current excitation, as shown in Fig. 16. Such sections are connected in cascade and the nodes are renumbered according to the decomposition tree. User's program EXAMP12 and the corresponding results obtained are shown on pp. 90-92.

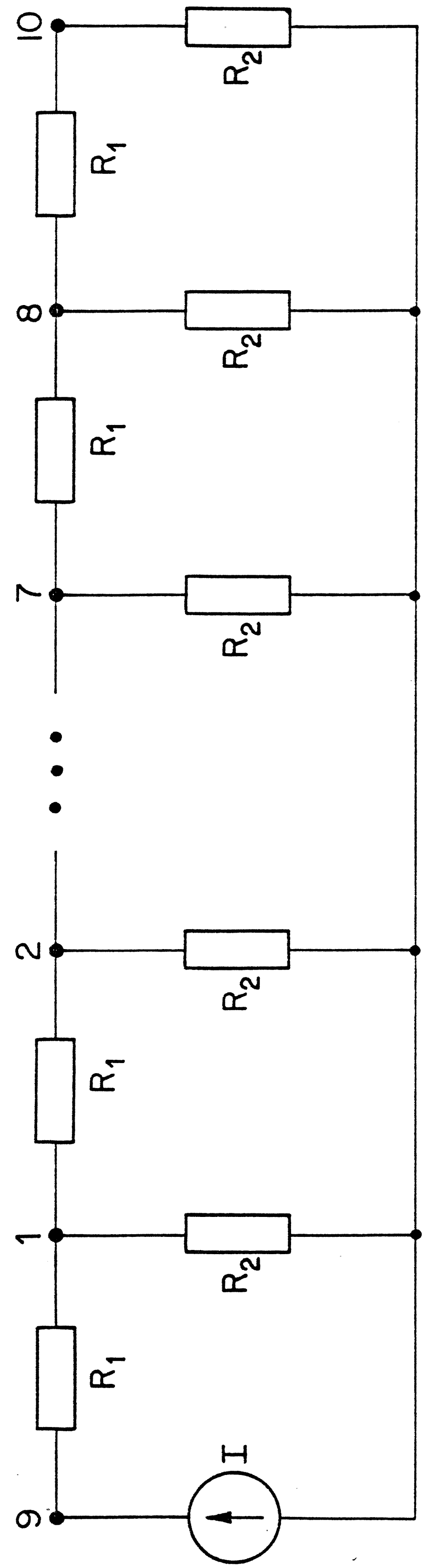


Fig. 16 Proper block for Example 12.

```
      PROGRAM LADDER(INPUT,OUTPUT,TAPE4=INPUT,TAPE6=OUTPUT,TAPE2)
C
C         THIS PROGRAM GENERATES DATA FOR THE DECOMPOSED SYSTEM
C
C         OF LINEAR EQUATIONS WHICH ARE THE NODAL EQUATIONS OF
C
C         A LADDER NETWORK
C
      INTEGER INTWA(2000)
      COMPLEX CMPLXWA(10000)
C
C         READ SIZE OF THE NETWORK
C
      WRITE (6,10)
   10 FORMAT (1H ,/," NUMBER OF DECOMPOSITION LEVELS:",11X,/," NUMBER OF
     1 NODES IN PROPER BLOCK:",/," LENGTH OF DATA RECORD:",/)
      READ*,LEVEL,NN,IN25
      CALL SYSTEM (LEVEL,NN,IREC2,NALL,IN5,IN9,IN17,IN18,IN19,IN25,IN26,
     1INTWA,CMPLXWA)
      WRITE (6,20) IREC2,NALL,IN5,IN9,IN17,IN18,IN19,IN26
   20 FORMAT (,/," NUMBER OF DATA RECORDS + 1 =",I6,/," NUMBER OF ALL NO
     1DES =",I6,/," NUMBER OF ALL PROPER BLOCKS =",I6,/," THE HIGHEST IN
     2DEX OF ALL THE SUBGRAPHS =",I6,/," MAXIMUM AREA FOR SPARSE MATRIX
     3=",I6,/," NUMBER OF NONZERO ELEMENTS IN MAX SUBMATRIX =",I6,,/," T
     4HE LOWEST INDEX OVER ALL DECOMPOSITION NODES =",I6,,/," NUMBER OF
     5ALL DECOMPOSITION NODES =",I4)
      STOP
      END
C
C
      SUBROUTINE SYSTEM (LEVEL,NN,IREC2,NALL,IN5,IN9,IN17,IN18,IN19,IN25
     1,IN26,INTWA,CMPLXWA)
      INTEGER INTWA(1)
      COMPLEX CMPLXWA(1)
      IN9=2**(LEVEL+1)-1
      ISBS=IN9/2
      IN5=ISBS+1
      NALL=IN5*NN-ISBS
      IN18=3*NN-2
      IN17=IN18
      IN19=NALL-ISBS+1
      IN26=ISBS
      IREC2=(2*IN5+IN5*(2+(3*NN-2)*2+NN+(NN-1)/2+1))/IN25+5
      IA1=1
      IA2=IA1+2*IN5
      IA3=IA2+2*IN5
      IA4=IA3+IREC2
      IA5=IA4+IN18
      IA6=IA5+IN18
      IA7=IA6+(NN-1)/2+1
   10 FORMAT (1H ,/," LENGTH OF WORKING AREA INTWA MUST BE NOT LESS THAN
     1",I6,/)
      WRITE (6,10) IA7
      IB1=1
      IB2=1+2+2*IN18+NN+(NN-1)/2+1
      IB3=IB2+IN25
      IB4=IB3+IN18
      IB5=IB4+NN
      WRITE (6,20) IB5
   20 FORMAT (1H ,/," LENGTH OF WORKING AREA CMPLXWA MUST BE NOT LESS TH
     1AN",I7/)
      CALL SYSGEN (INTWA(IA1),INTWA(IA2),INTWA(IA3),CMPLXWA(IB1),CMPLXWA
     1(IB2),IREC2,CMPLXWA(IB3),INTWA(IA4),INTWA(IA5),CMPLXWA(IB4),INTWA(
     2IA6),NN,IN5,IN9,IN25,ISBS,NALL,IN17)
      RETURN
```

```
      END
C
C
      SUBROUTINE SYSGEN (NSUB,NASUB,INDEX2,CE,DREC,IREC2,AK,NROW,NCOL,V,
     1NON,NN,IN5,IN9,IN25,ISBS,NALL,IN17)
      INTEGER NSUB(1),NASUB(1),INDEX2(1),NROW(1),NCOL(1),NON(1)
      COMPLEX CE(1),DREC(1),AK(1),V(1)
C
C         THIS SUBROUTINE GENERATES A SYSTEM OF LINEAR EQUATIONS
C
C         AND STORES IT IN THE RANDOM ACCESS MULTI RECORD FILE
C
C         NO.2
C
      CALL OPENMS (2,INDEX2,IREC2,0)
      IND2=1
      LDREC=2*IN25
C
C           WRITE DIRECTORY
C
      NB=1
      NC=2+(3*NN-2)*2+NN+(NN-1)/2+1
      ND=2*IN5+1
      DO 20 I=1,IN5
      ND1=ND+NC
      IF (ND1.LE.IN25) GO TO 10
      NB=NB+1
      ND1=NC+1
      ND=1
   10 DREC(I)=CMPLX(FLOAT(I+ISBS),FLOAT(NB))
      DREC(I+IN5)=CMPLX(FLOAT(NC),FLOAT(ND))
      ND=ND1
   20 CONTINUE
      MMNS=2*IN5
      DO 30 I=1,MMNS
      NSUB(I)=INT(REAL(DREC(I)))
      NASUB(I)=INT(AIMAG(DREC(I)))
   30 CONTINUE
      NINT=NN-2
      NEL=3*NN-2
      LES=IN17
      CE(2)=CMPLX(FLOAT(LES),FLOAT(NEL))
      DO 90 NBS=1,IN5
      CALL BLOCK (NBS,NALL,IN9,AK,NROW,NCOL,V,NON,NSUB,NN,LRI)
      IF (LRI.EQ.0) CE(1)=CMPLX(FLOAT(NN),FLOAT(NINT))
      IF (LRI.EQ.1) CE(1)=CMPLX(FLOAT(NN),FLOAT(NINT+1))
      DO 40 J=1,NEL
      CE(2+J)=AK(J)
      CE(2+NEL+J)=CMPLX(FLOAT(NROW(J)),FLOAT(NCOL(J)))
   40 CONTINUE
      NEL2=2*NEL+2
      DO 50 J=1,NN
      CE(NEL2+J)=V(J)
   50 CONTINUE
      NEL3=NEL2+NN
      NEL4=(NN-1)/2+1
      DO 60 J=1,NEL4
      CE(NEL3+J)=CMPLX(FLOAT(NON(2*J-1)),FLOAT(NON(2*J)))
   60 CONTINUE
C
C         STORE A SUBNETWORK IN THE DATA RECORD
C
      IF (NASUB(NBS).EQ.IND2) GO TO 70
      CALL WRITMS (2,DREC,LDREC,IND2)
      IND2=NASUB(NBS)
```

```
   70 IAD=NASUB(NBS+IN5)-1
      IG=NSUB(NBS+IN5)
      DO 80 I=1,IG
      DREC(IAD+I)=CE(I)
   80 CONTINUE
   90 CONTINUE
C
C          STORE FINAL RESULTS
C
      CALL WRITMS (2,DREC,LDREC,IND2)
      RETURN
      END
C
C
      SUBROUTINE BLOCK(NBS,NALL,IN9,AK,NROW,NCOL,V,NON,NSUB,NN,LRI)
      INTEGER NROW(1),NCOL(1),NON(1),NSUB(1)
      COMPLEX AK(1),V(1)
C
C          THIS SUBROUTINE GENERATES DATA DESCRIBING ONE SUBGRAPH
C
      LRI=0
      DO 1 I=1,NN
      AK(I)=CMPLX(3.,0.)
      NROW(I)=I
      NCOL(I)=I
      NROW(NN+I)=I+1
      NCOL(NN+I)=I
      AK(NN+I)=CMPLX(-1.,0.)
    1 V(I)=CMPLX(0.,0.)
      AK(NN-1)=CMPLX(1.,0.)
      AK(NN)=CMPLX(2.,0.)
      NN2=2*NN-1
      NCOL(NN2)=NN-2
      NCOL(NN2-1)=1
      NN1=NN-1
      DO 2 I=1,NN1
      NROW(NN2+I)=I
      NCOL(NN2+I)=I+1
    2 AK(NN2+I)=CMPLX(-1.,0.)
      NN3=NN2+NN-1
      NROW(NN3)=NN-2
      NROW(NN3-1)=1
C
C          CALCULATE ORIGINAL INDICES OF THE NODES
C
      NX0=NSUB(NBS)
      NXR=NX0+1
      NXL=NX0-1
      NX0=NX0/2
      NXR=NXR/2
      NXL=NXL/2
      IF(NX0.EQ.NXR)GOTO 11
      IF(NX0.EQ.NXL)GOTO 4
   11 IF(NXL.EQ.0) GOTO5
      NX0=NX0/2
      NXL=NXL/2
      IF(NX0.EQ.NXL) GOTO 3
      GOTO 11
    4 NX0=NX0/2
      NXR=NXR/2
      IF(NX0.NE.NXR) GO TO 4
      IF(NX0.EQ.0) GOTO 6
    3 NL=NALL-NXL+1
      NR=NALL-NXR+1
      GOTO 7
```

```
C
C         FIRST SECTION
C
    5 NL=1
      NR=NALL-NXR+1
      LRI=1
      GOTO 7
C
C         LAST SECTION
C
    6 NR=NALL-IN9/2
      NL=NALL-NXL+1
      LRI=1
    7 IN1=2+(NSUB(NBS)-IN9/2-1)*(NN-2)
      IN2=IN1+NN-3
      J=0
      DO 8 I=IN1,IN2
      J=J+1
    8 NON(J)=I
      J=J+1
      NON(J)=NL
      V(J)=CMPLX(1.,0.)
      J=J+1
      NON(J)=NR
      J=J+1
      NON(J)=0
C
C         RENUMBER LAST TWO NODES
C
      IF(NL.LT.NR)RETURN
      AK(NN-1)=CMPLX(2.,0.)
      AK(NN)=CMPLX(1.,0.)
      NROW(NN2)=NN-1
      NROW(NN2-1)=NN
      NCOL(NN3-1)=NN
      NCOL(NN3)=NN-1
      NON(J-1)=NL
      NON(J-2)=NR
      V(J-2)=CMPLX(0.,0.)
      V(J-1)=CMPLX(1.,0.)
      RETURN
      END
```

```
      PROGRAM EXAMP12(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 12
C
      DIMENSION IN(33), IM(5), INTWA(229)
      COMPLEX CMPLXWA(2900),SOLR(200)
      DATA IN/0,0,0,4,4,0,3,6,7,4,1229,10,0,0,1,10,28,28,35,5,0,0,0,0,15
     100,3,0,1,1,7,6,5,4/
      DATA IM/1,0,229,2900,100/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 12",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,20) IA3,IN(5),IN(15),IN(12),IA1
   20 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
   30 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

EXAMPLE 12

LADDER NETWORK WITH 37 NODES DECOMPOSED INTO 4 PROPER BLOCKS
NUMBER OF BASIC RECORDS 1
LENGTH OF ADDRESSING RECORDS 10
NUMBER OF SOLUTION RECORDS 1

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
37	.447368E+00	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	VALUE IMAGINARY
36	.447571E+00	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
35	.447291E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
2	.618314E+00	0.	5	.380829E-01	0.	8	.671785E-01	0.
3	.236628E+00	0.	6	.226784E-01	0.	9	.171583E+00	0.
4	.915704E-01	0.	7	.299523E-01	0.	1	.161831E+01	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE	VARIABLE	VALUE	VARIABLE	VALUE

	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
10	.171130E+00	0.	13	.131622E-01	0.	16	.657897E-01	0.
11	.658190E-01	0.	14	.131595E-01	0.	17	.171053E+00	0.
12	.263271E-01	0.	15	.263164E-01	0.			

IN SUBGRAPH NO 6

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
18	.171053E+00	0.	21	.131573E-01	0.	24	.657782E-01	0.
19	.657894E-01	0.	22	.131562E-01	0.	25	.171023E+00	0.
20	.263156E-01	0.	23	.263115E-01	0.			

IN SUBGRAPH NO 7

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
26	.170850E+00	0.	29	.952138E-02	0.	32	.534909E-03	0.
27	.652589E-01	0.	30	.363738E-02	0.	33	.213964E-03	0.
28	.249268E-01	0.	31	.139076E-02	0.	34	.106982E-03	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 37

CPU TIME: .253 SECONDS

Example 13

After the first execution of subroutine CSDSLE1 with IN(8) = 6, the user may preserve the results of the LR decomposition of the partitioned system stored in files Nos. 1, 3 and 5 and execute CSDSLE1 with an updated coefficient matrix and RHS vector in one or more proper blocks. Before execution of the user's main program EXAMP13, the program EXAMP6, as described in Example 6, was executed and local files 1, 3 and 5 preserved. Then the data file No. 2 containing information about the updated proper block G_5 was created by execution of the user's program DECSYS5 listed on p. 94. Coefficient (2,1) in G_5 has been assigned the value 3. This corresponds to an equivalent change in the coefficient (8,5) of the matrix from p. 11. The solution of the new system is listed on p. 96.

```
      PROGRAM DECSYS5(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(2)
      COMPLEX CE(11)
C
C         DIRECTORY
C
      DATA CE/(5.,1.),(9.,3.),
C
C         SUBGRAPH G5
C
     +(3.,1.),(9.,1.),(3.,0.),(2.,1.),
     +(0.,0.),(0.,0.),(9.,0.),
     +(5.,8.),(9.,0.)/
C
C         IREC2    THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                  NOT LESS THAN IN(20)+1
C
      IREC2=2
      CALL OPENMS(2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS(2,CE,22,1)
      STOP
      END
```

```
      PROGRAM EXAMP13( INPUT,OUTPUT,RESULT,TAPE4= INPUT,TAPE6=RESULT,
     1TAPE1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 13
C
      DIMENSION IN(33), IM(5), INTWA(137)
      COMPLEX CMPLXWA(640),SOLR(20)
      DATA IN/0,0,0,4,4,0,3,7,7,1,540,10,0,0,1,4,12,12,7,1,0,0,1,0,27,3
     1,0,1,1,7,6,5,4/
      DATA IM/1,0,137,640,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 13",/)
      CALL CSDSLE1 ( IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 13

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	IMAGINARY
9	.301358E+01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	IMAGINARY
8	.250000E+00	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	IMAGINARY
7	-.167256E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	IMAGINARY
6	.500000E+00	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE REAL	IMAGINARY
5	-.117119E+01	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	IMAGINARY	VARIABLE	VALUE REAL	IMAGINARY
3	.233737E+00	0.	4	-.355969E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	IMAGINARY	VARIABLE	VALUE REAL	IMAGINARY
1	-.157092E+00	0.	2	.290544E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 9

CPU TIME: .164 SECONDS

Example 14

To investigate the savings in computer time, a ladder network of medium size was generated by the program LADDER described in Example 12 with LEVEL = 4, NN = 50 and IN25 = 1500. The only difference was that ungrounded resistors have the values $.5.10^{-3}$. Generated file No. 2 contains data about the ladder network with 785 nodes decomposed into 16 proper blocks with IREC2 = 15, NALL = 785, IN5 = 16, IN9 = 16, IN17 = 148, IN18 = 148, IN19 = 771 and IN26 = 15 (see Example 12).

This network was analyzed first with the help of the user's main program EXAMP14, listed on p. 99, and local files 1, 3 and 5 were preserved. The results are listed on pp. 102-112. Subsequently, the user's main program EXAM14A was executed.

In EXAM14A, the complex vector CE contains one data record describing the updated coefficient matrix corresponding to the proper block G_{26}, in which the values of coefficients (2,1) and (1,2) have been altered from -2000 to -1. The solution of the altered system is listed on pp. 112-122.

As seen from the results, the change in the solution vector is the largest in subgraph G_{26}, where the coefficients have been altered, becoming smaller for the variables associated with nodes far away from the disturbed area. They can be barely detected in subgraphs G_{22} and G_{30} and are not noticed in G_{16}-G_{21} and G_{31}, at least for the assumed 6 digits accuracy.

It should be noted that this ladder is very sensitive for changes of elements and excitations because of the very low resistance between the nodes compared with shunt resistances. A change in the solution can also be observed when the solution is calculated for the subgraph G_{26}

and all its ascendants. To obtain this result the user's main program EXAM14B was executed instead of EXAM14A. In EXAM14B the indicator IN(22)=2, which means that only the solution for the subgraphs specified by the user will be calculated. In this case, G_{26} is altered, therefore IN(58)=26, as required.

By inspecting the results listed on pp. 123-124 we can see how the changes in the subnetwork coefficients affect the solution at partition nodes of all the ascendants of a given subgraph as well as at all internal nodes of this subgraph. This type of analysis is recommended particularly for extremely large networks, because the changes can be localized to a certain area and the updated approximate solution can be obtained much faster than for the whole network.

```
      PROGRAM EXAMP14(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 14
C
      DIMENSION IN(57), IM(5), INTWA(1063)
      COMPLEX CMPLXWA(7000),SOLR(200)
      DATA IN/0,0,0,16,16,0,3,6,31,16,4400,32,0,0,20,50,148,148,771,9,0
     1,0,0,0,1500,15,0,1,1,1,2,2,1,1,2,2,2,2,2,2,1,16*(0)/
      DATA IM/1,0,1063,7000,100/
C
      DO 10 I=1,16
      IN(41+I)=32-I
   10 CONTINUE
      WRITE (6,20)
   20 FORMAT (1H ,/," EXAMPLE 14",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,30) IA3,IN(5),IN(15),IN(12),IA1
   30 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,40) CPU
   40 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

```
      PROGRAM EXAM14A(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 14A
C
      DIMENSION IN(57), IM(5), INTWA(1200)
      COMPLEX CMPLXWA(7000),SOLR(200),CE(58)
      DATA CE/(26.,1.),(56.,3.),(50.,48.),(148.,2.),(-1.,0.),(-1.,0.),(2
     1.,1.),(1.,2.),50*((0.,0.))/
      DATA IN/0,0,0,16,16,0,3,7,31,1,4400,32,0,0,20,50,148,148,771,9,0,0
     1,1,0,1500,15,0,1,1,1,2,2,1,1,2,2,2,2,2,2,1,16*(0)/
      DATA IM/2,0,1200,7000,100/
C
      DO 10 I=1,16
      IN(41+I)=32-I
   10 CONTINUE
      IA=4*IN(17)+3*IN(16)+IN(11)
      DO 20 I=1,50
      CE(8+I)=CMPLX(0.,0.)
   20 CONTINUE
      CE(57)=CMPLX(1.,0.)
      DO 30 I=1,58
      CMPLXWA(IA+I)=CE(I)
   30 CONTINUE
      WRITE (6,40)
   40 FORMAT (1H ,/," EXAMPLE 14A",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,50) IA3,IN(5),IN(15),IN(12),IA1
   50 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,60) CPU
   60 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

```
      PROGRAM EXAM14B(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C          THIS IS THE MAIN PROGRAM FOR EXAMPLE 14B
C
      DIMENSION IN(58), IM(5), INTWA(1200)
      COMPLEX CMPLXWA(7000),SOLR(200),CE(58)
      DATA CE/(26.,1.),(56.,3.),(50.,48.),(148.,2.),(-1.,0.),(-1.,0.),(2
     1.,1.),(1.,2.),50*((0.,0.))/
      DATA IN/0,0,0,1,16,0,3,7,31,1,4400,32,0,0,20,50,148,148,771,9,0,2,
     11,0,1500,15,0,1,1,1,2,2,1,1,2,2,2,2,2,2,1,17*(0)/
      DATA IM/2,0,1200,7000,100/
C
      DO 10 I=1,16
      IN(41+I)=32-I
   10 CONTINUE
      IN(58)=26
      IA=4*IN(17)+3*IN(16)+IN(11)
      DO 20 I=1,50
      CE(8+I)=CMPLX(0.,0.)
   20 CONTINUE
      CE(57)=CMPLX(1.,0.)
      DO 30 I=1,58
      CMPLXWA(IA+I)=CE(I)
   30 CONTINUE
      WRITE (6,40)
   40 FORMAT (1H ,/," EXAMPLE 14B",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,50) IA3,IN(5),IN(15),IN(12),IA1
   50 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,60) CPU
   60 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

```
EXAMPLE 14

LADDER NETWORK WITH   785 NODES DECOMPOSED INTO    16 PROPER BLOCKS
NUMBER OF BASIC RECORDS    20
LENGTH OF ADDRESSING RECORDS    32
NUMBER OF SOLUTION RECORDS     8

MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 6791


SOLUTION
--------

IN SUBGRAPH NO   1

 VARIABLE                VALUE

                 REAL           IMAGINARY

   785       .376167E-01   0.

IN SUBGRAPH NO   2

 VARIABLE                VALUE

                 REAL           IMAGINARY

   784       .376167E-01   0.

IN SUBGRAPH NO   3

 VARIABLE                VALUE

                 REAL           IMAGINARY

   783       .376166E-01   0.

IN SUBGRAPH NO   4

 VARIABLE                VALUE

                 REAL           IMAGINARY

   782       .376539E-01   0.

IN SUBGRAPH NO   5
```

```
VARIABLE                VALUE
                REAL            IMAGINARY
  781       .376167E-01    0.
IN SUBGRAPH NO    6
VARIABLE                VALUE
                REAL            IMAGINARY
  780       .376167E-01    0.
IN SUBGRAPH NO    7
VARIABLE                VALUE
                REAL            IMAGINARY
  779       .375820E-01    0.
IN SUBGRAPH NO    8
VARIABLE                VALUE
                REAL            IMAGINARY
  778       .388063E-01    0.
IN SUBGRAPH NO    9
VARIABLE                VALUE
                REAL            IMAGINARY
  777       .376178E-01    0.
IN SUBGRAPH NO   10
VARIABLE                VALUE
                REAL            IMAGINARY
  776       .376167E-01    0.
IN SUBGRAPH NO   11
VARIABLE                VALUE
```

	REAL	IMAGINARY
775	.376167E-01	0.

IN SUBGRAPH NO 12

VARIABLE	VALUE REAL	IMAGINARY
774	.376167E-01	0.

IN SUBGRAPH NO 13

VARIABLE	VALUE REAL	IMAGINARY
773	.376167E-01	0.

IN SUBGRAPH NO 14

VARIABLE	VALUE REAL	IMAGINARY
772	.376156E-01	0.

IN SUBGRAPH NO 15

VARIABLE	VALUE REAL	IMAGINARY
771	.365083E-01	0.

IN SUBGRAPH NO 16

VARIABLE	VALUE REAL	IMAGINARY	VARIABLE	VALUE REAL	IMAGINARY	VARIABLE	VALUE REAL	IMAGINARY
2	.706219E-01	0.	19	.249393E-01	0.	36	.198292E-01	0.
3	.659750E-01	0.	20	.238139E-01	0.	37	.203942E-01	0.
4	.616530E-01	0.	21	.228075E-01	0.	38	.210611E-01	0.
5	.576493E-01	0.	22	.219151E-01	0.	39	.218334E-01	0.
6	.539288E-01	0.	23	.211323E-01	0.	40	.227148E-01	0.
7	.504780E-01	0.	24	.204552E-01	0.	41	.237098E-01	0.
8	.472795E-01	0.	25	.198803E-01	0.	42	.248233E-01	0.
9	.443175E-01	0.	26	.194049E-01	0.	43	.260610E-01	0.

10	.415770E-01	0.	27	.190264E-01	0.	44	.274289E-01	0.
11	.390444E-01	0.	28	.187431E-01	0.	45	.289340E-01	0.
12	.367071E-01	0.	29	.185536E-01	0.	46	.305838E-01	0.
13	.345533E-01	0.	30	.184567E-01	0.	47	.323865E-01	0.
14	.325722E-01	0.	31	.184522E-01	0.	48	.343511E-01	0.
15	.307540E-01	0.	32	.185399E-01	0.	49	.364875E-01	0.
16	.290896E-01	0.	33	.187204E-01	0.	1	.756219E-01	0.
17	.275706E-01	0.	34	.189944E-01	0.			
18	.261895E-01	0.	35	.193634E-01	0.			

IN SUBGRAPH NO 17

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
50	.363191E-01	0.	66	.151211E-01	0.	82	.154230E-01	0.
51	.340136E-01	0.	67	.146264E-01	0.	83	.160358E-01	0.
52	.318781E-01	0.	68	.142049E-01	0.	84	.167287E-01	0.
53	.299020E-01	0.	69	.138543E-01	0.	85	.175053E-01	0.
54	.280754E-01	0.	70	.135731E-01	0.	86	.183694E-01	0.
55	.263892E-01	0.	71	.133596E-01	0.	87	.193254E-01	0.
56	.248349E-01	0.	72	.132130E-01	0.	88	.203780E-01	0.
57	.234048E-01	0.	73	.131325E-01	0.	89	.215325E-01	0.
58	.220917E-01	0.	74	.131176E-01	0.	90	.227946E-01	0.
59	.208891E-01	0.	75	.131683E-01	0.	91	.241707E-01	0.
60	.197910E-01	0.	76	.132849E-01	0.	92	.256677E-01	0.
61	.187918E-01	0.	77	.134678E-01	0.	93	.272930E-01	0.
62	.178865E-01	0.	78	.137181E-01	0.	94	.290548E-01	0.
63	.170707E-01	0.	79	.140370E-01	0.	95	.309618E-01	0.
64	.163402E-01	0.	80	.144261E-01	0.	96	.330237E-01	0.
65	.156915E-01	0.	81	.148874E-01	0.	97	.352507E-01	0.

IN SUBGRAPH NO 18

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
98	.352454E-01	0.	114	.147747E-01	0.	130	.153112E-01	0.
99	.330131E-01	0.	115	.143036E-01	0.	131	.159316E-01	0.
100	.309459E-01	0.	116	.139041E-01	0.	132	.166317E-01	0.
101	.290335E-01	0.	117	.135741E-01	0.	133	.174149E-01	0.
102	.272661E-01	0.	118	.133119E-01	0.	134	.182852E-01	0.
103	.256352E-01	0.	119	.131164E-01	0.	135	.192469E-01	0.
104	.241324E-01	0.	120	.129864E-01	0.	136	.203048E-01	0.
105	.227502E-01	0.	121	.129213E-01	0.	137	.214643E-01	0.
106	.214818E-01	0.	122	.129208E-01	0.	138	.227311E-01	0.
107	.203208E-01	0.	123	.129850E-01	0.	139	.241116E-01	0.
108	.192615E-01	0.	124	.131140E-01	0.	140	.256126E-01	0.

109	.182984E-01	0.	125	.133086E-01	0.	141	.272417E-01	0.
110	.174268E-01	0.	126	.135698E-01	0.	142	.290069E-01	0.
111	.166424E-01	0.	127	.138938E-01	0.	143	.309173E-01	0.
112	.159411E-01	0.	128	.142974E-01	0.	144	.329822E-01	0.
113	.153196E-01	0.	129	.147674E-01	0.	145	.352120E-01	0.

IN SUBGRAPH NO 19

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
146	.352118E-01	0.	162	.147638E-01	0.	178	.153077E-01	0.
147	.329818E-01	0.	163	.142935E-01	0.	179	.159284E-01	0.
148	.309168E-01	0.	164	.138947E-01	0.	180	.166286E-01	0.
149	.290063E-01	0.	165	.135653E-01	0.	181	.174121E-01	0.
150	.272408E-01	0.	166	.133038E-01	0.	182	.182825E-01	0.
151	.256116E-01	0.	167	.131087E-01	0.	183	.192444E-01	0.
152	.241104E-01	0.	168	.129793E-01	0.	184	.203026E-01	0.
153	.227297E-01	0.	169	.129147E-01	0.	185	.214622E-01	0.
154	.214627E-01	0.	170	.129147E-01	0.	186	.227291E-01	0.
155	.203031E-01	0.	171	.129792E-01	0.	187	.241097E-01	0.
156	.192449E-01	0.	172	.131087E-01	0.	188	.256109E-01	0.
157	.182830E-01	0.	173	.133037E-01	0.	189	.272400E-01	0.
158	.174124E-01	0.	174	.135652E-01	0.	190	.290054E-01	0.
159	.166290E-01	0.	175	.138945E-01	0.	191	.309159E-01	0.
160	.159287E-01	0.	176	.142933E-01	0.	192	.329809E-01	0.
161	.153080E-01	0.	177	.147636E-01	0.	193	.352108E-01	0.

IN SUBGRAPH NO 20

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
194	.352108E-01	0.	210	.147635E-01	0.	226	.153076E-01	0.
195	.329808E-01	0.	211	.142932E-01	0.	227	.159283E-01	0.
196	.309158E-01	0.	212	.138944E-01	0.	228	.166285E-01	0.
197	.290054E-01	0.	213	.135650E-01	0.	229	.174120E-01	0.
198	.272400E-01	0.	214	.133035E-01	0.	230	.182825E-01	0.
199	.256108E-01	0.	215	.131085E-01	0.	231	.192444E-01	0.
200	.241097E-01	0.	216	.129790E-01	0.	232	.203025E-01	0.
201	.227291E-01	0.	217	.129145E-01	0.	233	.214621E-01	0.
202	.214621E-01	0.	218	.129145E-01	0.	234	.227291E-01	0.
203	.203025E-01	0.	219	.129790E-01	0.	235	.241097E-01	0.
204	.192444E-01	0.	220	.131085E-01	0.	236	.256108E-01	0.
205	.182825E-01	0.	221	.133035E-01	0.	237	.272400E-01	0.
206	.174120E-01	0.	222	.135650E-01	0.	238	.290054E-01	0.
207	.166286E-01	0.	223	.138944E-01	0.	239	.309158E-01	0.
208	.159283E-01	0.	224	.142932E-01	0.	240	.329808E-01	0.

209	.153076E-01	0.	225	.147635E-01	0.	241	.352107E-01	0.

IN SUBGRAPH NO 21

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
242	.352107E-01	0.	258	.147635E-01	0.	274	.153076E-01	0.
243	.329808E-01	0.	259	.142932E-01	0.	275	.159283E-01	0.
244	.309158E-01	0.	260	.138944E-01	0.	276	.166285E-01	0.
245	.290054E-01	0.	261	.135650E-01	0.	277	.174120E-01	0.
246	.272400E-01	0.	262	.133035E-01	0.	278	.182825E-01	0.
247	.256108E-01	0.	263	.131085E-01	0.	279	.192444E-01	0.
248	.241097E-01	0.	264	.129790E-01	0.	280	.203025E-01	0.
249	.227291E-01	0.	265	.129145E-01	0.	281	.214621E-01	0.
250	.214621E-01	0.	266	.129145E-01	0.	282	.227291E-01	0.
251	.203025E-01	0.	267	.129790E-01	0.	283	.241097E-01	0.
252	.192444E-01	0.	268	.131085E-01	0.	284	.256108E-01	0.
253	.182825E-01	0.	269	.133035E-01	0.	285	.272400E-01	0.
254	.174120E-01	0.	270	.135650E-01	0.	286	.290054E-01	0.
255	.166285E-01	0.	271	.138944E-01	0.	287	.309158E-01	0.
256	.159283E-01	0.	272	.142932E-01	0.	288	.329808E-01	0.
257	.153076E-01	0.	273	.147635E-01	0.	289	.352107E-01	0.

IN SUBGRAPH NO 22

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
290	.352107E-01	0.	306	.147635E-01	0.	322	.153076E-01	0.
291	.329808E-01	0.	307	.142932E-01	0.	323	.159283E-01	0.
292	.309158E-01	0.	308	.138944E-01	0.	324	.166285E-01	0.
293	.290054E-01	0.	309	.135650E-01	0.	325	.174120E-01	0.
294	.272400E-01	0.	310	.133035E-01	0.	326	.182825E-01	0.
295	.256108E-01	0.	311	.131085E-01	0.	327	.192444E-01	0.
296	.241097E-01	0.	312	.129790E-01	0.	328	.203025E-01	0.
297	.227291E-01	0.	313	.129145E-01	0.	329	.214621E-01	0.
298	.214621E-01	0.	314	.129145E-01	0.	330	.227291E-01	0.
299	.203025E-01	0.	315	.129790E-01	0.	331	.241097E-01	0.
300	.192444E-01	0.	316	.131085E-01	0.	332	.256108E-01	0.
301	.182825E-01	0.	317	.133035E-01	0.	333	.272400E-01	0.
302	.174120E-01	0.	318	.135650E-01	0.	334	.290054E-01	0.
303	.166285E-01	0.	319	.138944E-01	0.	335	.309158E-01	0.
304	.159283E-01	0.	320	.142932E-01	0.	336	.329808E-01	0.
305	.153076E-01	0.	321	.147635E-01	0.	337	.352107E-01	0.

IN SUBGRAPH NO 23

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
338	.352107E-01	0.	354	.147635E-01	0.	370	.153076E-01	0.
339	.329808E-01	0.	355	.142932E-01	0.	371	.159283E-01	0.
340	.309158E-01	0.	356	.138944E-01	0.	372	.166285E-01	0.
341	.290054E-01	0.	357	.135650E-01	0.	373	.174120E-01	0.
342	.272400E-01	0.	358	.133035E-01	0.	374	.182825E-01	0.
343	.256108E-01	0.	359	.131085E-01	0.	375	.192444E-01	0.
344	.241097E-01	0.	360	.129790E-01	0.	376	.203025E-01	0.
345	.227291E-01	0.	361	.129145E-01	0.	377	.214621E-01	0.
346	.214621E-01	0.	362	.129145E-01	0.	378	.227291E-01	0.
347	.203025E-01	0.	363	.129790E-01	0.	379	.241097E-01	0.
348	.192444E-01	0.	364	.131085E-01	0.	380	.256108E-01	0.
349	.182825E-01	0.	365	.133035E-01	0.	381	.272400E-01	0.
350	.174120E-01	0.	366	.135650E-01	0.	382	.290054E-01	0.
351	.166285E-01	0.	367	.138944E-01	0.	383	.309158E-01	0.
352	.159283E-01	0.	368	.142932E-01	0.	384	.329808E-01	0.
353	.153076E-01	0.	369	.147635E-01	0.	335	.352107E-01	0.

IN SUBGRAPH NO 24

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
386	.352107E-01	0.	402	.147635E-01	0.	418	.153076E-01	0.
387	.329808E-01	0.	403	.142932E-01	0.	419	.159283E-01	0.
388	.309158E-01	0.	404	.138944E-01	0.	420	.166285E-01	0.
389	.290054E-01	0.	405	.135650E-01	0.	421	.174120E-01	0.
390	.272400E-01	0.	406	.133035E-01	0.	422	.182825E-01	0.
391	.256108E-01	0.	407	.131085E-01	0.	423	.192444E-01	0.
392	.241097E-01	0.	408	.129790E-01	0.	424	.203025E-01	0.
393	.227291E-01	0.	409	.129145E-01	0.	425	.214621E-01	0.
394	.214621E-01	0.	410	.129145E-01	0.	426	.227291E-01	0.
395	.203025F-01	0.	411	.129790E-01	0.	427	.241097E-01	0.
396	.192444E-01	0.	412	.131085E-01	0.	428	.256108E-01	0.
397	.182825E-01	0.	413	.133035E-01	0.	429	.272400E-01	0.
398	.174120E-01	0.	414	.135650E-01	0.	430	.290054E-01	0.
399	.166285E-01	0.	415	.138944E-01	0.	431	.309158E-01	0.
400	.159283E-01	0.	416	.142932E-01	0.	432	.329808E-01	0.
401	.153076E-01	0.	417	.147635E-01	0.	433	.352107E-01	0.

IN SUBGRAPH NO 25

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY

434	.352107E-01	0.	450	.147635E-01	0.	466	.153076E-01	0.
435	.329808E-01	0.	451	.142932E-01	0.	467	.159283E-01	0.
436	.309158E-01	0.	452	.138944E-01	0.	468	.166285E-01	0.
437	.290054E-01	0.	453	.135650E-01	0.	469	.174120E-01	0.
438	.272400E-01	0.	454	.133035E-01	0.	470	.182825E-01	0.
439	.256108E-01	0.	455	.131085E-01	0.	471	.192444E-01	0.
440	.241097E-01	0.	456	.129790E-01	0.	472	.203025E-01	0.
441	.227291E-01	0.	457	.129145E-01	0.	473	.214621E-01	0.
442	.214621E-01	0.	458	.129145E-01	0.	474	.227291E-01	0.
443	.203025E-01	0.	459	.129790E-01	0.	475	.241097E-01	0.
444	.192444E-01	0.	460	.131085E-01	0.	476	.256108E-01	0.
445	.182825E-01	0.	461	.133035E-01	0.	477	.272400E-01	0.
446	.174120E-01	0.	462	.135650E-01	0.	478	.290054E-01	0.
447	.166285E-01	0.	463	.138944E-01	0.	479	.309158E-01	0.
448	.159283E-01	0.	464	.142932E-01	0.	480	.329808E-01	0.
449	.153076E-01	0.	465	.147635E-01	0.	481	.352107E-01	0.

IN SUBGRAPH NO 26

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
482	.352107E-01	0.	498	.147635E-01	0.	514	.153076E-01	0.
483	.329808E-01	0.	499	.142932E-01	0.	515	.159283E-01	0.
484	.309158E-01	0.	500	.138944E-01	0.	516	.166285E-01	0.
485	.290054E-01	0.	501	.135650E-01	0.	517	.174120E-01	0.
486	.272400E-01	0.	502	.133035E-01	0.	518	.182825E-01	0.
487	.256108E-01	0.	503	.131085E-01	0.	519	.192444E-01	0.
488	.241097E-01	0.	504	.129790E-01	0.	520	.203025E-01	0.
489	.227291E-01	0.	505	.129145E-01	0.	521	.214621E-01	0.
490	.214621E-01	0.	506	.129145E-01	0.	522	.227291E-01	0.
491	.203025E-01	0.	507	.129790E-01	0.	523	.241097E-01	0.
492	.192444E-01	0.	508	.131085E-01	0.	524	.256108E-01	0.
493	.182825E-01	0.	509	.133035E-01	0.	525	.272400E-01	0.
494	.174120E-01	0.	510	.135650E-01	0.	526	.290054E-01	0.
495	.166285E-01	0.	511	.138944E-01	0.	527	.309158E-01	0.
496	.159283E-01	0.	512	.142932E-01	0.	528	.329808E-01	0.
497	.153076E-01	0.	513	.147635E-01	0.	529	.352107E-01	0.

IN SUBGRAPH NO 27

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
530	.352107E-01	0.	546	.147635E-01	0.	562	.153076E-01	0.
531	.329808E-01	0.	547	.142932E-01	0.	563	.159282E-01	0.
532	.309158E-01	0.	548	.138944E-01	0.	564	.166285E-01	0.
533	.290054E-01	0.	549	.135650E-01	0.	565	.174120E-01	0.

534	.272400E-01	0.	550	.133035E-01	0.	566	.182824E-01	0.
535	.256108E-01	0.	551	.131085E-01	0.	567	.192443E-01	0.
536	.241097E-01	0.	552	.129790E-01	0.	568	.203025E-01	0.
537	.227291E-01	0.	553	.129144E-01	0.	569	.214621E-01	0.
538	.214621E-01	0.	554	.129144E-01	0.	570	.227290E-01	0.
539	.203025E-01	0.	555	.129790E-01	0.	571	.241096E-01	0.
540	.192444E-01	0.	556	.131085E-01	0.	572	.256108E-01	0.
541	.182825E-01	0.	557	.133035E-01	0.	573	.272400E-01	0.
542	.174120E-01	0.	558	.135650E-01	0.	574	.290054E-01	0.
543	.166285E-01	0.	559	.138944E-01	0.	575	.309158E-01	0.
544	.159282E-01	0.	560	.142932E-01	0.	576	.329808E-01	0.
545	.153076E-01	0.	561	.147635E-01	0.	577	.352107E-01	0.

IN SUBGRAPH NO 28

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
578	.352107E-01	0.	594	.147634E-01	0.	610	.153073E-01	0.
579	.329808E-01	0.	595	.142931E-01	0.	611	.159279E-01	0.
580	.309158E-01	0.	596	.138943E-01	0.	612	.166281E-01	0.
581	.290053E-01	0.	597	.135649E-01	0.	613	.174115E-01	0.
582	.272399E-01	0.	598	.133034E-01	0.	614	.182820E-01	0.
583	.256108E-01	0.	599	.131083E-01	0.	615	.192439E-01	0.
584	.241096E-01	0.	600	.129789E-01	0.	616	.203020E-01	0.
585	.227290E-01	0.	601	.129143E-01	0.	617	.214615E-01	0.
586	.214621E-01	0.	602	.129143E-01	0.	618	.227285E-01	0.
587	.203024E-01	0.	603	.129788E-01	0.	619	.241090E-01	0.
588	.192443E-01	0.	604	.131083E-01	0.	620	.256101E-01	0.
589	.182824E-01	0.	605	.133033E-01	0.	621	.272392E-01	0.
590	.174119E-01	0.	606	.135648E-01	0.	622	.290046E-01	0.
591	.166284E-01	0.	607	.138941E-01	0.	623	.309149E-01	0.
592	.159282E-01	0.	608	.142929E-01	0.	624	.329799E-01	0.
593	.153075E-01	0.	609	.147632E-01	0.	625	.352097E-01	0.

IN SUBGRAPH NO 29

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
626	.352096E-01	0.	642	.147599E-01	0.	658	.152964E-01	0.
627	.329796E-01	0.	643	.142893E-01	0.	659	.159162E-01	0.
628	.309145E-01	0.	644	.138902E-01	0.	660	.166156E-01	0.
629	.290040E-01	0.	645	.135606E-01	0.	661	.173981E-01	0.
630	.272385E-01	0.	646	.132987E-01	0.	662	.182676E-01	0.
631	.256091E-01	0.	647	.131033E-01	0.	663	.192284E-01	0.
632	.241079E-01	0.	648	.129735E-01	0.	664	.202854E-01	0.
633	.227272E-01	0.	649	.129085E-01	0.	665	.214438E-01	0.

634	.214601E-01	0.	650	.129081E-01	0.	666	.227094E-01	0.
635	.203003E-01	0.	651	.129722E-01	0.	667	.240885E-01	0.
636	.192420E-01	0.	652	.131012E-01	0.	668	.255881E-01	0.
637	.182799E-01	0.	653	.132956E-01	0.	669	.272156E-01	0.
638	.174092E-01	0.	654	.135566E-01	0.	670	.289792E-01	0.
639	.166256E-01	0.	655	.138853E-01	0.	671	.308877E-01	0.
640	.159251E-01	0.	656	.142835E-01	0.	672	.329507E-01	0.
641	.153042E-01	0.	657	.147531E-01	0.	673	.351784E-01	0.

IN SUBGRAPH NO 30

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
674	.351735E-01	0.	690	.146481E-01	0.	706	.149500E-01	0.
675	.329468E-01	0.	691	.141694E-01	0.	707	.155444E-01	0.
676	.303729E-01	0.	692	.137615E-01	0.	708	.162166E-01	0.
677	.289594E-01	0.	693	.134224E-01	0.	709	.169698E-01	0.
678	.271906E-01	0.	694	.131504E-01	0.	710	.178079E-01	0.
679	.255578E-01	0.	695	.129442E-01	0.	711	.187351E-01	0.
680	.240528E-01	0.	696	.128027E-01	0.	712	.197559E-01	0.
681	.226680E-01	0.	697	.127252E-01	0.	713	.208755E-01	0.
682	.213966E-01	0.	698	.127113E-01	0.	714	.220995E-01	0.
683	.202321E-01	0.	699	.127610E-01	0.	715	.234339E-01	0.
684	.191689E-01	0.	700	.128745E-01	0.	716	.248856E-01	0.
685	.182014E-01	0.	701	.130523E-01	0.	717	.264616E-01	0.
686	.173250E-01	0.	702	.132955E-01	0.	718	.281700E-01	0.
687	.165352E-01	0.	703	.136051E-01	0.	719	.300192E-01	0.
688	.158281E-01	0.	704	.139827E-01	0.	720	.320185E-01	0.
689	.152001E-01	0.	705	.144303E-01	0.	721	.341780E-01	0.

IN SUBGRAPH NO 31

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
722	.340211E-01	0.	739	.103363E-01	0.	756	.346719E-02	0.
723	.317041E-01	0.	740	.964765E-02	0.	757	.328649E-02	0.
724	.295455E-01	0.	741	.900721E-02	0.	758	.312222E-02	0.
725	.275347E-01	0.	742	.841180E-02	0.	759	.297356E-02	0.
726	.256616E-01	0.	743	.785845E-02	0.	760	.283977E-02	0.
727	.239168E-01	0.	744	.734439E-02	0.	761	.272017E-02	0.
728	.222916E-01	0.	745	.686705E-02	0.	762	.261418E-02	0.
729	.207778E-01	0.	746	.642405E-02	0.	763	.252126E-02	0.
730	.193679E-01	0.	747	.601317E-02	0.	764	.244095E-02	0.
731	.180548E-01	0.	748	.563235E-02	0.	765	.237284E-02	0.
732	.168321E-01	0.	749	.527970E-02	0.	766	.231659E-02	0.
733	.156934E-01	0.	750	.495344E-02	0.	767	.227193E-02	0.

734	.146333E-01	0.	751	.465195E-02	0.	768	.223862E-02	0.
735	.136463E-01	0.	752	.437373E-02	0.	769	.221651E-02	0.
736	.127276E-01	0.	753	.411737E-02	0.	770	.220549E-02	0.
737	.118725E-01	0.	754	.388159E-02	0.			
738	.110767E-01	0.	755	.366523E-02	0.			

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 785

CPU TIME: 2.740 SECONDS

EXAMPLE 14A

LADDER NETWORK WITH 785 NODES DECOMPOSED INTO 16 PROPER BLOCKS
NUMBER OF BASIC RECORDS 20
LENGTH OF ADDRESSING RECORDS 32
NUMBER OF SOLUTION RECORDS 8

MAXIMUM INTEGER AREA REQUIRED IS EQUAL TO 1063

MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 6791

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	IMAGINARY
785	.375801E-01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	IMAGINARY
784	.376167E-01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	IMAGINARY
783	.386843E-01	0.

```
IN SUBGRAPH NO   4
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   782       .376539E-01   0.
IN SUBGRAPH NO   5
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   781       .376166E-01   0.
IN SUBGRAPH NO   6
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   780       .282707E-03   0.
IN SUBGRAPH NO   7
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   779       .375830E-01   0.
IN SUBGRAPH NO   8
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   778       .388063E-01   0.
IN SUBGRAPH NO   9
 VARIABLE                 VALUE
                 REAL             IMAGINARY
   777       .376178E-01   0.
IN SUBGRAPH NO  10
```

```
VARIABLE              VALUE
               REAL            IMAGINARY
  776       .376167E-01    0.
IN SUBGRAPH NO  11
VARIABLE              VALUE
               REAL            IMAGINARY
  775       .376155E-01    0.
IN SUBGRAPH NO  12
VARIABLE              VALUE
               REAL            IMAGINARY
  774       .364481E-01    0.
IN SUBGRAPH NO  13
VARIABLE              VALUE
               REAL            IMAGINARY
  773       .717272E-01    0.
IN SUBGRAPH NO  14
VARIABLE              VALUE
               REAL            IMAGINARY
  772       .376490E-01    0.
IN SUBGRAPH NO  15
VARIABLE              VALUE
               REAL            IMAGINARY
  771       .365083E-01    0.
IN SUBGRAPH NO  16
VARIABLE              VALUE          VARIABLE          VALUE          VARIABLE          VALUE
```

	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
2	.706219E-01	0.	19	.249393E-01	0.	36	.198292E-01	0.
3	.659750E-01	0.	20	.238139E-01	0.	37	.203942E-01	0.
4	.616580E-01	0.	21	.228075E-01	0.	38	.210611E-01	0.
5	.576493E-01	0.	22	.219151E-01	0.	39	.218334E-01	0.
6	.539288E-01	0.	23	.211323E-01	0.	40	.227148E-01	0.
7	.504780E-01	0.	24	.204552E-01	0.	41	.237098E-01	0.
8	.472795E-01	0.	25	.198803E-01	0.	42	.248233E-01	0.
9	.443175E-01	0.	26	.194049E-01	0.	43	.260610E-01	0.
10	.415770E-01	0.	27	.190264E-01	0.	44	.274289E-01	0.
11	.390444E-01	0.	28	.187431E-01	0.	45	.289340E-01	0.
12	.367071E-01	0.	29	.185536E-01	0.	46	.305838E-01	0.
13	.345533E-01	0.	30	.184567E-01	0.	47	.323865E-01	0.
14	.325722E-01	0.	31	.184522E-01	0.	48	.343511E-01	0.
15	.307540E-01	0.	32	.185399E-01	0.	49	.364875E-01	0.
16	.290896E-01	0.	33	.187204E-01	0.	1	.756219E-01	0.
17	.275706E-01	0.	34	.189944E-01	0.			
18	.261895E-01	0.	35	.193634E-01	0.			

IN SUBGRAPH NO 17

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
50	.363191E-01	0.	66	.151211E-01	0.	82	.154230E-01	0.
51	.340136E-01	0.	67	.146264E-01	0.	83	.160358E-01	0.
52	.318781E-01	0.	68	.142049E-01	0.	84	.167287E-01	0.
53	.299020E-01	0.	69	.138543E-01	0.	85	.175053E-01	0.
54	.280754E-01	0.	70	.135731E-01	0.	86	.183694E-01	0.
55	.263892E-01	0.	71	.133596E-01	0.	87	.193254E-01	0.
56	.248349E-01	0.	72	.132130E-01	0.	88	.203780E-01	0.
57	.234048E-01	0.	73	.131325E-01	0.	89	.215325E-01	0.
58	.220917E-01	0.	74	.131176E-01	0.	90	.227946E-01	0.
59	.208891E-01	0.	75	.131683E-01	0.	91	.241707E-01	0.
60	.197910E-01	0.	76	.132849E-01	0.	92	.256677E-01	0.
61	.187918E-01	0.	77	.134678E-01	0.	93	.272930E-01	0.
62	.178865E-01	0.	78	.137181E-01	0.	94	.290548E-01	0.
63	.170707E-01	0.	79	.140370E-01	0.	95	.309618E-01	0.
64	.163402E-01	0.	80	.144261E-01	0.	96	.330237E-01	0.
65	.156915E-01	0.	81	.148874E-01	0.	97	.352507E-01	0.

IN SUBGRAPH NO 18

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY

98	.352454E-01	0.	114	.147747E-01	0.	130	.153112E-01	0.
99	.330131E-01	0.	115	.143036E-01	0.	131	.159316E-01	0.
100	.309459E-01	0.	116	.139041E-01	0.	132	.166317E-01	0.
101	.290335E-01	0.	117	.135741E-01	0.	133	.174149E-01	0.
102	.272661E-01	0.	118	.133119E-01	0.	134	.182852E-01	0.
103	.256352E-01	0.	119	.131164E-01	0.	135	.192469E-01	0.
104	.241324E-01	0.	120	.129864E-01	0.	136	.203048E-01	0.
105	.227502E-01	0.	121	.129213E-01	0.	137	.214643E-01	0.
106	.214818E-01	0.	122	.129208E-01	0.	138	.227311E-01	0.
107	.203208E-01	0.	123	.129850E-01	0.	139	.241116E-01	0.
108	.192615E-01	0.	124	.131140E-01	0.	140	.256126E-01	0.
109	.182984E-01	0.	125	.133086E-01	0.	141	.272417E-01	0.
110	.174268E-01	0.	126	.135698E-01	0.	142	.290069E-01	0.
111	.166424E-01	0.	127	.138988E-01	0.	143	.309173E-01	0.
112	.159411E-01	0.	128	.142974E-01	0.	144	.329822E-01	0.
113	.153196E-01	0.	129	.147674E-01	0.	145	.352120E-01	0.

IN SUBGRAPH NO 19

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
146	.352118E-01	0.	162	.147638E-01	0.	178	.153077E-01	0.
147	.329818E-01	0.	163	.142935E-01	0.	179	.159284E-01	0.
148	.309168E-01	0.	164	.138947E-01	0.	180	.166286E-01	0.
149	.290063E-01	0.	165	.135653E-01	0.	181	.174121E-01	0.
150	.272408E-01	0.	166	.133038E-01	0.	182	.182825E-01	0.
151	.256116E-01	0.	167	.131087E-01	0.	183	.192444E-01	0.
152	.241104E-01	0.	168	.129793E-01	0.	184	.203026E-01	0.
153	.227297E-01	0.	169	.129147E-01	0.	185	.214622E-01	0.
154	.214627E-01	0.	170	.129147E-01	0.	186	.227291E-01	0.
155	.203031E-01	0.	171	.129792E-01	0.	187	.241097E-01	0.
156	.192449E-01	0.	172	.131087E-01	0.	188	.256109E-01	0.
157	.182830E-01	0.	173	.133037E-01	0.	189	.272400E-01	0.
158	.174124E-01	0.	174	.135652E-01	0.	190	.290054E-01	0.
159	.166290E-01	0.	175	.138945E-01	0.	191	.309159E-01	0.
160	.159287E-01	0.	176	.142933E-01	0.	192	.329809E-01	0.
161	.153080E-01	0.	177	.147636E-01	0.	193	.352108E-01	0.

IN SUBGRAPH NO 20

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
194	.352108E-01	0.	210	.147635E-01	0.	226	.153076E-01	0.
195	.329808E-01	0.	211	.142932E-01	0.	227	.159283E-01	0.
196	.309158E-01	0.	212	.138944E-01	0.	228	.166285E-01	0.
197	.290054E-01	0.	213	.135650E-01	0.	229	.174120E-01	0.

198	.272400E-01	0.	214	.133035E-01	0.	230	.182825E-01	0.
199	.256108E-01	0.	215	.131085E-01	0.	231	.192444E-01	0.
200	.241097E-01	0.	216	.129790E-01	0.	232	.203025E-01	0.
201	.227291E-01	0.	217	.129145E-01	0.	233	.214621E-01	0.
202	.214621E-01	0.	218	.129145E-01	0.	234	.227291E-01	0.
203	.203025E-01	0.	219	.129790E-01	0.	235	.241097E-01	0.
204	.192444E-01	0.	220	.131085E-01	0.	236	.256108E-01	0.
205	.182825E-01	0.	221	.133035E-01	0.	237	.272400E-01	0.
206	.174120E-01	0.	222	.135650E-01	0.	238	.290054E-01	0.
207	.166286E-01	0.	223	.138944E-01	0.	239	.309158E-01	0.
208	.159283E-01	0.	224	.142932E-01	0.	240	.329808E-01	0.
209	.153076E-01	0.	225	.147635E-01	0.	241	.352107E-01	0.

IN SUBGRAPH NO 21

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
242	.352107E-01	0.	258	.147635E-01	0.	274	.153076E-01	0.
243	.329808E-01	0.	259	.142932E-01	0.	275	.159282E-01	0.
244	.309158E-01	0.	260	.138944E-01	0.	276	.166285E-01	0.
245	.290054E-01	0.	261	.135650E-01	0.	277	.174120E-01	0.
246	.272400E-01	0.	262	.133035E-01	0.	278	.182824E-01	0.
247	.256108E-01	0.	263	.131085E-01	0.	279	.192443E-01	0.
248	.241097E-01	0.	264	.129790E-01	0.	280	.203025E-01	0.
249	.227291E-01	0.	265	.129144E-01	0.	281	.214621E-01	0.
250	.214621E-01	0.	266	.129144E-01	0.	282	.227290E-01	0.
251	.203025E-01	0.	267	.129790E-01	0.	283	.241096E-01	0.
252	.192444E-01	0.	268	.131085E-01	0.	284	.256108E-01	0.
253	.182825E-01	0.	269	.133035E-01	0.	285	.272400E-01	0.
254	.174120E-01	0.	270	.135650E-01	0.	286	.290054E-01	0.
255	.166285E-01	0.	271	.138944E-01	0.	287	.309158E-01	0.
256	.159282E-01	0.	272	.142932E-01	0.	288	.329808E-01	0.
257	.153076E-01	0.	273	.147635E-01	0.	289	.352107E-01	0.

IN SUBGRAPH NO 22

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
290	.352107E-01	0.	306	.147634E-01	0.	322	.153072E-01	0.
291	.329808E-01	0.	307	.142931E-01	0.	323	.159279E-01	0.
292	.309158E-01	0.	308	.138942E-01	0.	324	.166281E-01	0.
293	.290053E-01	0.	309	.135649E-01	0.	325	.174115E-01	0.
294	.272399E-01	0.	310	.133033E-01	0.	326	.182820E-01	0.
295	.256107E-01	0.	311	.131083E-01	0.	327	.192438E-01	0.
296	.241096E-01	0.	312	.129783E-01	0.	328	.203019E-01	0.
297	.227290E-01	0.	313	.129143E-01	0.	329	.214615E-01	0.

298	.214621E-01	0.	314	.129142E-01	0.	330	.227284E-01	0.
299	.203024E-01	0.	315	.129788E-01	0.	331	.241090E-01	0.
300	.192443E-01	0.	316	.131083E-01	0.	332	.256101E-01	0.
301	.182824E-01	0.	317	.133032E-01	0.	333	.272392E-01	0.
302	.174119E-01	0.	318	.135648E-01	0.	334	.290045E-01	0.
303	.166284E-01	0.	319	.138941E-01	0.	335	.309149E-01	0.
304	.159281E-01	0.	320	.142929E-01	0.	336	.329798E-01	0.
305	.153075E-01	0.	321	.147632E-01	0.	337	.352097E-01	0.

IN SUBGRAPH NO 23

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
338	.352095E-01	0.	354	.147597E-01	0.	370	.152958E-01	0.
339	.329795E-01	0.	355	.142891E-01	0.	371	.159156E-01	0.
340	.309144E-01	0.	356	.138960E-01	0.	372	.166149E-01	0.
341	.290039E-01	0.	357	.135603E-01	0.	373	.173974E-01	0.
342	.272384E-01	0.	358	.132985E-01	0.	374	.182668E-01	0.
343	.256091E-01	0.	359	.131031E-01	0.	375	.192276E-01	0.
344	.241078E-01	0.	360	.129732E-01	0.	376	.202844E-01	0.
345	.227271E-01	0.	361	.129082E-01	0.	377	.214428E-01	0.
346	.214600E-01	0.	362	.129077E-01	0.	378	.227083E-01	0.
347	.203002E-01	0.	363	.129718E-01	0.	379	.240874E-01	0.
348	.192419E-01	0.	364	.131008E-01	0.	380	.255869E-01	0.
349	.182798E-01	0.	365	.132952E-01	0.	381	.272143E-01	0.
350	.174091E-01	0.	366	.135561E-01	0.	382	.289778E-01	0.
351	.166255E-01	0.	367	.138848E-01	0.	383	.308862E-01	0.
352	.159249E-01	0.	368	.142830E-01	0.	384	.329491E-01	0.
353	.153041E-01	0.	369	.147525E-01	0.	385	.351766E-01	0.

IN SUBGRAPH NO 24

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
386	.351715E-01	0.	402	.146418E-01	0.	418	.149305E-01	0.
387	.329387E-01	0.	403	.141626E-01	0.	419	.155236E-01	0.
388	.308706E-01	0.	404	.137542E-01	0.	420	.161942E-01	0.
389	.289569E-01	0.	405	.134146E-01	0.	421	.169458E-01	0.
390	.271879E-01	0.	406	.131421E-01	0.	422	.177822E-01	0.
391	.255549E-01	0.	407	.129352E-01	0.	423	.187074E-01	0.
392	.240497E-01	0.	408	.127931E-01	0.	424	.197262E-01	0.
393	.226647E-01	0.	409	.127149E-01	0.	425	.208436E-01	0.
394	.213930E-01	0.	410	.127093E-01	0.	426	.220653E-01	0.
395	.202283E-01	0.	411	.127491E-01	0.	427	.233972E-01	0.
396	.191648E-01	0.	412	.128618E-01	0.	428	.248462E-01	0.
397	.181970E-01	0.	413	.130387E-01	0.	429	.264194E-01	0.

398	.173203E-01	0.	414	.132808E-01	0.	430	.281246E-01	0.
399	.165301E-01	0.	415	.135894E-01	0.	431	.299705E-01	0.
400	.158226E-01	0.	416	.139659E-01	0.	432	.319663E-01	0.
401	.151942E-01	0.	417	.144122E-01	0.	433	.341219E-01	0.

IN SUBGRAPH NO 25

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
434	.339565E-01	0.	450	.108765E-01	0.	466	.326118E-02	0.
435	.316347E-01	0.	451	.101215E-01	0.	467	.299936E-02	0.
436	.294711E-01	0.	452	.941706E-02	0.	468	.275255E-02	0.
437	.274549E-01	0.	453	.875972E-02	0.	469	.251949E-02	0.
438	.255759E-01	0.	454	.814618E-02	0.	470	.229904E-02	0.
439	.238248E-01	0.	455	.757338E-02	0.	471	.209007E-02	0.
440	.221928E-01	0.	456	.703843E-02	0.	472	.189156E-02	0.
441	.206718E-01	0.	457	.653858E-02	0.	473	.170251E-02	0.
442	.192542E-01	0.	458	.607163E-02	0.	474	.152197E-02	0.
443	.179328E-01	0.	459	.563493E-02	0.	475	.134904E-02	0.
444	.167011E-01	0.	460	.522641E-02	0.	476	.118285E-02	0.
445	.155529E-01	0.	461	.484402E-02	0.	477	.102258E-02	0.
446	.144824E-01	0.	462	.448585E-02	0.	478	.867424E-03	0.
447	.134844E-01	0.	463	.415010E-02	0.	479	.716603E-03	0.
448	.125538E-01	0.	464	.383511E-02	0.	480	.569365E-03	0.
449	.116859E-01	0.	465	.353930E-02	0.	481	.424973E-03	0.

IN SUBGRAPH NO 26

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
482	.141854E-03	0.	498	.669808E-02	0.	514	.229190E-01	0.
483	.342029E-03	0.	499	.729886E-02	0.	515	.246335E-01	0.
484	.685059E-03	0.	500	.793614E-02	0.	516	.264711E-01	0.
485	.103151E-02	0.	501	.861310E-02	0.	517	.284411E-01	0.
486	.138313E-02	0.	502	.933312E-02	0.	518	.305533E-01	0.
487	.174166E-02	0.	503	.100998E-01	0.	519	.328183E-01	0.
488	.210889E-02	0.	504	.109170E-01	0.	520	.352473E-01	0.
489	.248667E-02	0.	505	.117888E-01	0.	521	.378526E-01	0.
490	.287689E-02	0.	506	.127195E-01	0.	522	.406472E-01	0.
491	.328149E-02	0.	507	.137138E-01	0.	523	.436449E-01	0.
492	.370249E-02	0.	508	.147767E-01	0.	524	.468609E-01	0.
493	.414201E-02	0.	509	.159134E-01	0.	525	.503113E-01	0.
494	.460224E-02	0.	510	.171298E-01	0.	526	.540131E-01	0.
495	.508548E-02	0.	511	.184318E-01	0.	527	.579851E-01	0.
496	.559415E-02	0.	512	.198259E-01	0.	528	.622469E-01	0.
497	.613079E-02	0.	513	.213192E-01	0.	529	.668200E-01	0.

IN SUBGRAPH NO 27

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
530	.669930E-01	0.	546	.250186E-01	0.	562	.186166E-01	0.
531	.625938E-01	0.	547	.238483E-01	0.	563	.190114E-01	0.
532	.585076E-01	0.	548	.227973E-01	0.	564	.195012E-01	0.
533	.547139E-01	0.	549	.218603E-01	0.	565	.200886E-01	0.
534	.511938E-01	0.	550	.210326E-01	0.	566	.207764E-01	0.
535	.479296E-01	0.	551	.203100E-01	0.	567	.215680E-01	0.
536	.449051E-01	0.	552	.196890E-01	0.	568	.224676E-01	0.
537	.421051E-01	0.	553	.191665E-01	0.	569	.234794E-01	0.
538	.395157E-01	0.	554	.187397E-01	0.	570	.246087E-01	0.
539	.371238E-01	0.	555	.184067E-01	0.	571	.258610E-01	0.
540	.349175E-01	0.	556	.181657E-01	0.	572	.272426E-01	0.
541	.328859E-01	0.	557	.180155E-01	0.	573	.287604E-01	0.
542	.310186E-01	0.	558	.179554E-01	0.	574	.304220E-01	0.
543	.293065E-01	0.	559	.179851E-01	0.	575	.322357E-01	0.
544	.277409E-01	0.	560	.181047E-01	0.	576	.342106E-01	0.
545	.263139E-01	0.	561	.183149E-01	0.	577	.363566E-01	0.

IN SUBGRAPH NO 28

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
578	.362055E-01	0.	594	.150844E-01	0.	610	.154108E-01	0.
579	.339077E-01	0.	595	.145922E-01	0.	611	.160244E-01	0.
580	.317794E-01	0.	596	.141729E-01	0.	612	.167181E-01	0.
581	.298101E-01	0.	597	.138245E-01	0.	613	.174953E-01	0.
582	.279897E-01	0.	598	.135453E-01	0.	614	.183601E-01	0.
583	.263094E-01	0.	599	.133337E-01	0.	615	.193166E-01	0.
584	.247605E-01	0.	600	.131889E-01	0.	616	.203697E-01	0.
585	.233355E-01	0.	601	.131100E-01	0.	617	.215247E-01	0.
586	.220272E-01	0.	602	.130966E-01	0.	618	.227873E-01	0.
587	.208289E-01	0.	603	.131487E-01	0.	619	.241638E-01	0.
588	.197349E-01	0.	604	.132666E-01	0.	620	.256612E-01	0.
589	.187395E-01	0.	605	.134507E-01	0.	621	.272868E-01	0.
590	.178378E-01	0.	606	.137022E-01	0.	622	.290489E-01	0.
591	.170253E-01	0.	607	.140221E-01	0.	623	.309563E-01	0.
592	.162979E-01	0.	608	.144122E-01	0.	624	.330184E-01	0.
593	.156520E-01	0.	609	.148743E-01	0.	625	.352456E-01	0.

IN SUBGRAPH NO 29

VARIABLE	VALUE	VARIABLE	VALUE	VARIABLE	VALUE

VARIABLE	REAL	IMAGINARY	VARIABLE	REAL	IMAGINARY	VARIABLE	REAL	IMAGINARY
626	.352407E-01	0.	642	.147699E-01	0.	658	.152997E-01	0.
627	.330086E-01	0.	643	.142987E-01	0.	659	.159193E-01	0.
628	.309415E-01	0.	644	.138989E-01	0.	660	.166185E-01	0.
629	.290291E-01	0.	645	.135687E-01	0.	661	.174008E-01	0.
630	.272619E-01	0.	646	.133063E-01	0.	662	.182700E-01	0.
631	.256310E-01	0.	647	.131104E-01	0.	663	.192307E-01	0.
632	.241283E-01	0.	648	.129801E-01	0.	664	.202875E-01	0.
633	.227461E-01	0.	649	.129147E-01	0.	665	.214457E-01	0.
634	.214778E-01	0.	650	.129138E-01	0.	666	.227112E-01	0.
635	.203168E-01	0.	651	.129775E-01	0.	667	.240902E-01	0.
636	.192574E-01	0.	652	.131061E-01	0.	668	.255897E-01	0.
637	.182942E-01	0.	653	.133003E-01	0.	669	.272171E-01	0.
638	.174226E-01	0.	654	.135609E-01	0.	670	.289806E-01	0.
639	.166380E-01	0.	655	.138893E-01	0.	671	.308890E-01	0.
640	.159367E-01	0.	656	.142872E-01	0.	672	.329519E-01	0.
641	.153150E-01	0.	657	.147565E-01	0.	673	.351795E-01	0.

IN SUBGRAPH NO 30

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
674	.351745E-01	0.	690	.146484E-01	0.	706	.149501E-01	0.
675	.329418E-01	0.	691	.141696E-01	0.	707	.155445E-01	0.
676	.308738E-01	0.	692	.137617E-01	0.	708	.162167E-01	0.
677	.289601E-01	0.	693	.134226E-01	0.	709	.169699E-01	0.
678	.271913E-01	0.	694	.131506E-01	0.	710	.178080E-01	0.
679	.255585E-01	0.	695	.129444E-01	0.	711	.187351E-01	0.
680	.240534E-01	0.	696	.128029E-01	0.	712	.197559E-01	0.
681	.226686E-01	0.	697	.127254E-01	0.	713	.208755E-01	0.
682	.213971E-01	0.	698	.127115E-01	0.	714	.220995E-01	0.
683	.202326E-01	0.	699	.127612E-01	0.	715	.234340E-01	0.
684	.191693E-01	0.	700	.128746E-01	0.	716	.248856E-01	0.
685	.182019E-01	0.	701	.130525E-01	0.	717	.264617E-01	0.
686	.173254E-01	0.	702	.132956E-01	0.	718	.281700E-01	0.
687	.165356E-01	0.	703	.136052E-01	0.	719	.300193E-01	0.
688	.158284E-01	0.	704	.139828E-01	0.	720	.320186E-01	0.
689	.152004E-01	0.	705	.144304E-01	0.	721	.341780E-01	0.

IN SUBGRAPH NO 31

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
722	.340211E-01	0.	739	.103363E-01	0.	756	.346719E-02	0.

723	.317041E-01	0.	740	.964766E-02	0.	757	.328649E-02	0.
724	.295456E-01	0.	741	.900722E-02	0.	758	.312222E-02	0.
725	.275348E-01	0.	742	.841181E-02	0.	759	.297356E-02	0.
726	.256616E-01	0.	743	.785845E-02	0.	760	.283977E-02	0.
727	.239168E-01	0.	744	.734439E-02	0.	761	.272018E-02	0.
728	.222916E-01	0.	745	.686706E-02	0.	762	.261418E-02	0.
729	.207778E-01	0.	746	.642405E-02	0.	763	.252126E-02	0.
730	.193679E-01	0.	747	.601317E-02	0.	764	.244095E-02	0.
731	.180549E-01	0.	748	.563236E-02	0.	765	.237284E-02	0.
732	.168321E-01	0.	749	.527970E-02	0.	766	.231659E-02	0.
733	.156935E-01	0.	750	.495345E-02	0.	767	.227193E-02	0.
734	.146333E-01	0.	751	.465196E-02	0.	768	.223863E-02	0.
735	.136463E-01	0.	752	.437373E-02	0.	769	.221652E-02	0.
736	.127276E-01	0.	753	.411737E-02	0.	770	.220549E-02	0.
737	.118725E-01	0.	754	.388160E-02	0.			
738	.110767E-01	0.	755	.366523E-02	0.			

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 785

CPU TIME: 1.480 SECONDS

EXAMPLE 14B

```
LADDER NETWORK WITH   770 NODES DECOMPOSED INTO    16 PROPER BLOCKS
NUMBER OF BASIC RECORDS    20
LENGTH OF ADDRESSING RECORDS    32
NUMBER OF SOLUTION RECORDS     8

MAXIMUM INTEGER AREA REQUIRED IS EQUAL TO 1063


MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 6791


SOLUTION
--------

IN SUBGRAPH NO   1

 VARIABLE              VALUE

                REAL          IMAGINARY

   783      .375801E-01   0.

IN SUBGRAPH NO   3

 VARIABLE              VALUE

                REAL          IMAGINARY

   783      .386843E-01   0.

IN SUBGRAPH NO   6

 VARIABLE              VALUE

                REAL          IMAGINARY

   789      .282707E-03   0.

IN SUBGRAPH NO  13

 VARIABLE              VALUE

                REAL          IMAGINARY

   773      .717272E-01   0.
```

IN SUBGRAPH NO 26

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
482	.141854E-03	0.	498	.669808E-02	0.	514	.229190E-01	0.
483	.342029E-03	0.	499	.729886E-02	0.	515	.246335E-01	0.
484	.685059E-03	0.	500	.793614E-02	0.	516	.264711E-01	0.
485	.103151E-02	0.	501	.861310E-02	0.	517	.284411E-01	0.
486	.138313E-02	0.	502	.933312E-02	0.	518	.305533E-01	0.
487	.174166E-02	0.	503	.100998E-01	0.	519	.328183E-01	0.
488	.210889E-02	0.	504	.109170E-01	0.	520	.352473E-01	0.
489	.248667E-02	0.	505	.117888E-01	0.	521	.378526E-01	0.
490	.287689E-02	0.	506	.127195E-01	0.	522	.406472E-01	0.
491	.328149E-02	0.	507	.137138E-01	0.	523	.436449E-01	0.
492	.370249E-02	0.	508	.147767E-01	0.	524	.468609E-01	0.
493	.414201E-02	0.	509	.159134E-01	0.	525	.503113E-01	0.
494	.460224E-02	0.	510	.171298E-01	0.	526	.540131E-01	0.
495	.508548E-02	0.	511	.184318E-01	0.	527	.579851E-01	0.
496	.559415E-02	0.	512	.198259E-01	0.	528	.622469E-01	0.
497	.613079E-02	0.	513	.213192E-01	0.	529	.668200E-01	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 52

CPU TIME: .340 SECONDS

Example 15

The user's main program EXAMP15 illustrates how to obtain a new solution for an updated RHS. After execution of program EXAMP6, as described in Example 6 files, Nos. 1, 3 and 5 were preserved. Then data file No. 2, containing information about the updated RHS vector in proper block G_5, was created by execution of the user's program DECSYS6 listed on p. 126. After that, the user's program EXAMP15 was executed. EXAMP15 and the corresponding results are listed on pp. 127-129.

```
      PROGRAM DECSYS6(TAPE2)
C
C         THIS PROGRAM CREATES THE RANDOM FILE CONTAINING INFORMATION
C
C         ABOUT THE SUBGRAPHS OF THE DECOMPOSED GRAPH
C
      INTEGER INDEX2(2)
      COMPLEX CE(5)
C
C         DIRECTORY
C
      DATA CE/(5.,1.),(3.,3.),
C
C         SUBGRAPH G5
C
     +(0.,0.),(0.,0.),(9.,0.)/
C
C         IREC2    THE NUMBER OF RECORDS IN RANDOM FILE NO.2. HAVE TO BE
C                  NOT LESS THAN IN(20)+1
C
      IREC2=2
      CALL OPENMS(2,INDEX2,IREC2,0)
C
C         STORE INFORMATION ABOUT SUBNETWORKS TO BE ANALYZED
C
      CALL WRITMS(2,CE,22,1)
      STOP
      END
```

```
      PROGRAM EXAMP15(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 15
C
      DIMENSION IN(33), IM(5), INTWA(137)
      COMPLEX CMPLXWA(716),SOLR(20)
      DATA IN/0,0,0,4,4,0,3,8,7,1,540,10,0,0,1,4,12,12,7,1,0,0,1,0,27,3,
     10,1,1,7,6,5,4/
      DATA IM/1,0,137,716,10/
C
      WRITE (6,10)
   10 FORMAT (1H ,/," EXAMPLE 15",/)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      STOP
      END
```

EXAMPLE 15

MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 640

SOLUTION

IN SUBGRAPH NO 1

VARIABLE	VALUE REAL	VALUE IMAGINARY
9	.345698E+01	0.

IN SUBGRAPH NO 2

VARIABLE	VALUE REAL	VALUE IMAGINARY
8	.123924E+01	0.

IN SUBGRAPH NO 3

VARIABLE	VALUE REAL	VALUE IMAGINARY
7	-.185777E+00	0.

IN SUBGRAPH NO 4

VARIABLE	VALUE REAL	VALUE IMAGINARY
6	-.147849E+01	0.

IN SUBGRAPH NO 5

VARIABLE	VALUE REAL	VALUE IMAGINARY
5	-.197849E+01	0.

IN SUBGRAPH NO 6

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
3	.279658E+00	0.	4	-.435464E+00	0.

IN SUBGRAPH NO 7

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY

```
     1     -.155505E+00   0.                  2      .292660E+00   0.
NUMBER OF VARIABLES IN THE SOLUTION VECTOR   9
CPU TIME:     .146 SECONDS
```

Example 16

The ladder network of Example 14 is used to demonstrate the effect of a change in the RHS vector in a subgraph of a bigger network. After execution of the user's main program EXAMP14, as described in Example 14 files Nos. 1, 3 and 5 were preserved and user's program EXAM16A was executed. In this program, the current excitation in the proper block G_{26} was given the value 20 instead of 1. The results obtained are listed on pp. 133-143. Again, it can be seen that the changes in the solution vector vanish with increasing distance from the new excitation and they cannot be observed for the assumed 6 digit accuracy in subgraphs G_{16}-G_{21} (compare with the results of Example 14).

Again, the user may restrict the solution to be calculated for certain subgraphs and their ascendants, to estimate the changes. The user's program EXAM16B, listed on p. 132, was used instead of EXAM16A to obtain the results listed on pp. 144-145.

```
      PROGRAM EXAM16A(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 16
C
      DIMENSION IN(57), IM(5), INTWA(1200)
      COMPLEX CMPLXWA(7000),SOLR(200),CE(52)
      DATA CE/(26.,1.),(50.,3.),50*((0.,0.))/
      DATA IN/0,0,0,16,16,0,1,8,31,1,4400,32,0,0,20,50,148,148,771,9,0,0
     1,1,0,1500,15,0,1,1,1,2,2,1,1,2,2,2,2,2,2,1,16*(0)/
      DATA IM/2,0,1200,7000,100/
C
      DO 10 I=1,16
      IN(41+I)=32-I
   10 CONTINUE
      IA=4*IN(17)+3*IN(16)+IN(11)
      DO 20 I=1,50
      CE(2+I)=CMPLX(0.,0.)
   20 CONTINUE
      CE(51)=CMPLX(20.,0.)
      DO 30 I=1,52
      CMPLXWA(IA+I)=CE(I)
   30 CONTINUE
      WRITE (6,40)
   40 FORMAT (1H ,/," EXAMPLE 16",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,50) IA3,IN(5),IN(15),IN(12),IA1
   50 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,60) CPU
   60 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

```
      PROGRAM EXAM16B(INPUT,OUTPUT,RESULT,TAPE4=INPUT,TAPE6=RESULT,TAPE1
     1,TAPE2,TAPE3,TAPE5,TAPE7)
C
C         THIS IS THE MAIN PROGRAM FOR EXAMPLE 16B
C
      DIMENSION IN(58), IM(5), INTWA(1200)
      COMPLEX CMPLXWA(7000),SOLR(200),CE(52)
      DATA CE/(26.,1.),(50.,3.),50*((0.,0.))/
      DATA IN/0,0,0,1,16,0,3,8,31,1,4400,32,0,0,20,50,148,148,771,9,0,2,
     11,0,1500,15,0,1,1,1,2,2,1,1,2,2,2,2,2,2,1,17*(0)/
      DATA IM/2,0,1200,7000,100/
C
      DO 10 I=1,16
      IN(41+I)=32-I
   10 CONTINUE
      IN(58)=26
      IA=4*IN(17)+3*IN(16)+IN(11)
      DO 20 I=1,50
      CE(2+I)=CMPLX(0.,0.)
   20 CONTINUE
      CE(51)=CMPLX(20.,0.)
      DO 30 I=1,52
      CMPLXWA(IA+I)=CE(I)
   30 CONTINUE
      WRITE (6,40)
   40 FORMAT (1H ,/," EXAMPLE 16B",/)
      IA1=(IN(19)+IN(26)-1)/IM(5)+1
      IA3=IN(19)+IN(4)-2
      WRITE (6,50) IA3,IN(5),IN(15),IN(12),IA1
   50 FORMAT (1H ,/," LADDER NETWORK WITH",I6," NODES DECOMPOSED INTO",I
     16," PROPER BLOCKS",/," NUMBER OF BASIC RECORDS",I6,/," LENGTH OF A
     2DDRESSING RECORDS",I6,/," NUMBER OF SOLUTION RECORDS",I6)
      CALL SECOND (TM1)
      CALL CSDSLE1 (IN,INTWA,CMPLXWA,IM,SOLR)
      IF (IN(7).NE.1) STOP
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,60) CPU
   60 FORMAT (1H ,/," CPU TIME:",F8.3" SECONDS",/)
      STOP
      END
```

```
EXAMPLE 16

LADDER NETWORK WITH    785 NODES DECOMPOSED INTO     16 PROPER BLOCKS
NUMBER OF BASIC RECORDS     20
LENGTH OF ADDRESSING RECORDS    32
NUMBER OF SOLUTION RECORDS     8

MAXIMUM INTEGER AREA REQUIRED IS EQUAL TO 1063


MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 6791


SOLUTION
--------


IN SUBGRAPH NO    1

 VARIABLE                VALUE

                REAL            IMAGINARY

   785      .376037E-01    0.

IN SUBGRAPH NO    2

 VARIABLE                VALUE

                REAL            IMAGINARY

   784      .376167E-01    0.

IN SUBGRAPH NO    3

 VARIABLE                VALUE

                REAL            IMAGINARY

   783      .597014E-01    0.

IN SUBGRAPH NO    4

 VARIABLE                VALUE

                REAL            IMAGINARY

   782      .376539E-01    0.
```

```
IN SUBGRAPH NO    5
 VARIABLE                VALUE
                 REAL           IMAGINARY
   781       .376167E-01    0.
IN SUBGRAPH NO    6
 VARIABLE                VALUE
                 REAL           IMAGINARY
   780       .244028E-01    0.
IN SUBGRAPH NO    7
 VARIABLE                VALUE
                 REAL           IMAGINARY
   779       .376036E-01    0.
IN SUBGRAPH NO    8
 VARIABLE                VALUE
                 REAL           IMAGINARY
   778       .388063E-01    0.
IN SUBGRAPH NO    9
 VARIABLE                VALUE
                 REAL           IMAGINARY
   777       .376178E-01    0.
IN SUBGRAPH NO   10
 VARIABLE                VALUE
                 REAL           IMAGINARY
   776       .376167E-01    0.
IN SUBGRAPH NO   11
```

VARIABLE	VALUE REAL	VALUE IMAGINARY
775	.376163E-01	0.

IN SUBGRAPH NO 12

VARIABLE	VALUE REAL	VALUE IMAGINARY
774	.372031E-01	0.

IN SUBGRAPH NO 13

VARIABLE	VALUE REAL	VALUE IMAGINARY
773	.743176E+00	0.

IN SUBGRAPH NO 14

VARIABLE	VALUE REAL	VALUE IMAGINARY
772	.383069E-01	0.

IN SUBGRAPH NO 15

VARIABLE	VALUE REAL	VALUE IMAGINARY
771	.365089E-01	0.

IN SUBGRAPH NO 16

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
2	.706219E-01	0.	19	.249393E-01	0.	36	.198292E-01	0.
3	.659750E-01	0.	20	.238139E-01	0.	37	.203942E-01	0.
4	.616580E-01	0.	21	.228075E-01	0.	38	.210611E-01	0.
5	.576493E-01	0.	22	.219151E-01	0.	39	.218334E-01	0.
6	.539285E-01	0.	23	.211323E-01	0.	40	.227148E-01	0.

7	.504780E-01	0.	24	.204552E-01	0.	41	.237098E-01	0.
8	.472795E-01	0.	25	.198803E-01	0.	42	.248233E-01	0.
9	.443175E-01	0.	26	.194049E-01	0.	43	.260610E-01	0.
10	.415770E-01	0.	27	.190264E-01	0.	44	.274289E-01	0.
11	.390444E-01	0.	28	.187431E-01	0.	45	.289340E-01	0.
12	.367071E-01	0.	29	.185536E-01	0.	46	.305838E-01	0.
13	.345533E-01	0.	30	.184567E-01	0.	47	.323865E-01	0.
14	.325722E-01	0.	31	.184522E-01	0.	48	.343511E-01	0.
15	.307540E-01	0.	32	.185399E-01	0.	49	.364875E-01	0.
16	.290896E-01	0.	33	.187204E-01	0.	1	.756219E-01	0.
17	.275706E-01	0.	34	.189944E-01	0.			
18	.261895E-01	0.	35	.193634E-01	0.			

IN SUBGRAPH NO 17

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
50	.363191E-01	0.	66	.151211E-01	0.	82	.154230E-01	0.
51	.340136E-01	0.	67	.146264E-01	0.	83	.160358E-01	0.
52	.318781E-01	0.	68	.142049E-01	0.	84	.167287E-01	0.
53	.299020E-01	0.	69	.138543E-01	0.	85	.175053E-01	0.
54	.280754E-01	0.	70	.135731E-01	0.	86	.183694E-01	0.
55	.263892E-01	0.	71	.133596E-01	0.	87	.193254E-01	0.
56	.248349E-01	0.	72	.132130E-01	0.	88	.203780E-01	0.
57	.234048E-01	0.	73	.131325E-01	0.	89	.215325E-01	0.
58	.220917E-01	0.	74	.131176E-01	0.	90	.227946E-01	0.
59	.208891E-01	0.	75	.131683E-01	0.	91	.241707E-01	0.
60	.197910E-01	0.	76	.132849E-01	0.	92	.256677E-01	0.
61	.187918E-01	0.	77	.134678E-01	0.	93	.272930E-01	0.
62	.178865E-01	0.	78	.137181E-01	0.	94	.290548E-01	0.
63	.170707E-01	0.	79	.140370E-01	0.	95	.309618E-01	0.
64	.163402E-01	0.	80	.144261E-01	0.	96	.330237E-01	0.
65	.156915E-01	0.	81	.148874E-01	0.	97	.352507E-01	0.

IN SUBGRAPH NO 18

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
98	.352454E-01	0.	114	.147747E-01	0.	130	.153112E-01	0.
99	.330131E-01	0.	115	.143036E-01	0.	131	.159316E-01	0.
100	.309459E-01	0.	116	.139041E-01	0.	132	.166317E-01	0.
101	.290335E-01	0.	117	.135741E-01	0.	133	.174149E-01	0.
102	.272661E-01	0.	118	.133119E-01	0.	134	.182852E-01	0.
103	.256352E-01	0.	119	.131164E-01	0.	135	.192469E-01	0.
104	.241324E-01	0.	120	.129864E-01	0.	136	.203048E-01	0.
105	.227502E-01	0.	121	.129213E-01	0.	137	.214643E-01	0.

106	.214818E-01	0.	122	.129208E-01	0.	138	.227311E-01	0.
107	.203208E-01	0.	123	.129850E-01	0.	139	.241116E-01	0.
108	.192615E-01	0.	124	.131140E-01	0.	140	.256126E-01	0.
109	.182984E-01	0.	125	.133086E-01	0.	141	.272417E-01	0.
110	.174268E-01	0.	126	.135698E-01	0.	142	.290069E-01	0.
111	.166424E-01	0.	127	.138988E-01	0.	143	.309173E-01	0.
112	.159411E-01	0.	128	.142974E-01	0.	144	.329822E-01	0.
113	.153196E-01	0.	129	.147674E-01	0.	145	.352120E-01	0.

IN SUBGRAPH NO 19

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
146	.352118E-01	0.	162	.147638E-01	0.	178	.153077E-01	0.
147	.329818E-01	0.	163	.142935E-01	0.	179	.159284E-01	0.
148	.309168E-01	0.	164	.138947E-01	0.	180	.166286E-01	0.
149	.290063E-01	0.	165	.135653E-01	0.	181	.174121E-01	0.
150	.272408E-01	0.	166	.133038E-01	0.	182	.182825E-01	0.
151	.256116E-01	0.	167	.131087E-01	0.	183	.192444E-01	0.
152	.241104E-01	0.	168	.129793E-01	0.	184	.203026E-01	0.
153	.227297E-01	0.	169	.129147E-01	0.	185	.214622E-01	0.
154	.214627E-01	0.	170	.129147E-01	0.	186	.227291E-01	0.
155	.203031E-01	0.	171	.129792E-01	0.	187	.241097E-01	0.
156	.192449E-01	0.	172	.131087E-01	0.	188	.256109E-01	0.
157	.182830E-01	0.	173	.133037E-01	0.	189	.272400E-01	0.
158	.174124E-01	0.	174	.135652E-01	0.	190	.290054E-01	0.
159	.166290E-01	0.	175	.138945E-01	0.	191	.309159E-01	0.
160	.159287E-01	0.	176	.142933E-01	0.	192	.329809E-01	0.
161	.153080E-01	0.	177	.147636E-01	0.	193	.352108E-01	0.

IN SUBGRAPH NO 20

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
194	.352108E-01	0.	210	.147635E-01	0.	226	.153076E-01	0.
195	.329808E-01	0.	211	.142932E-01	0.	227	.159283E-01	0.
196	.309158E-01	0.	212	.138944E-01	0.	228	.166285E-01	0.
197	.290054E-01	0.	213	.135650E-01	0.	229	.174120E-01	0.
198	.272400E-01	0.	214	.133035E-01	0.	230	.182825E-01	0.
199	.256108E-01	0.	215	.131085E-01	0.	231	.192444E-01	0.
200	.241097E-01	0.	216	.129790E-01	0.	232	.203025E-01	0.
201	.227291E-01	0.	217	.129145E-01	0.	233	.214621E-01	0.
202	.214621E-01	0.	218	.129145E-01	0.	234	.227291E-01	0.
203	.203025E-01	0.	219	.129790E-01	0.	235	.241097E-01	0.
204	.192444E-01	0.	220	.131085E-01	0.	236	.256108E-01	0.
205	.182825E-01	0.	221	.133035E-01	0.	237	.272400E-01	0.

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
206	.174120E-01	0.	222	.135650E-01	0.	238	.290054E-01	0.
207	.166286E-01	0.	223	.138944E-01	0.	239	.309158E-01	0.
208	.159283E-01	0.	224	.142932E-01	0.	240	.329808E-01	0.
209	.153076E-01	0.	225	.147635E-01	0.	241	.352107E-01	0.

IN SUBGRAPH NO 21

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
242	.352107E-01	0.	258	.147635E-01	0.	274	.153076E-01	0.
243	.329808E-01	0.	259	.142932E-01	0.	275	.159282E-01	0.
244	.309158E-01	0.	260	.138944E-01	0.	276	.166285E-01	0.
245	.290054E-01	0.	261	.135650E-01	0.	277	.174120E-01	0.
246	.272400E-01	0.	262	.133035E-01	0.	278	.182825E-01	0.
247	.256108E-01	0.	263	.131085E-01	0.	279	.192444E-01	0.
248	.241097E-01	0.	264	.129790E-01	0.	280	.203025E-01	0.
249	.227291E-01	0.	265	.129145E-01	0.	281	.214621E-01	0.
250	.214621E-01	0.	266	.129145E-01	0.	282	.227291E-01	0.
251	.203025E-01	0.	267	.129790E-01	0.	283	.241097E-01	0.
252	.192444E-01	0.	268	.131085E-01	0.	284	.256108E-01	0.
253	.182825E-01	0.	269	.133035E-01	0.	285	.272400E-01	0.
254	.174120E-01	0.	270	.135650E-01	0.	286	.290054E-01	0.
255	.166285E-01	0.	271	.138944E-01	0.	287	.309158E-01	0.
256	.159283E-01	0.	272	.142932E-01	0.	288	.329808E-01	0.
257	.153076E-01	0.	273	.147635E-01	0.	289	.352107E-01	0.

IN SUBGRAPH NO 22

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
290	.352107E-01	0.	306	.147635E-01	0.	322	.153075E-01	0.
291	.329808E-01	0.	307	.142932E-01	0.	323	.159281E-01	0.
292	.309158E-01	0.	308	.138943E-01	0.	324	.166284E-01	0.
293	.290054E-01	0.	309	.135650E-01	0.	325	.174118E-01	0.
294	.272400E-01	0.	310	.133034E-01	0.	326	.182823E-01	0.
295	.256108E-01	0.	311	.131084E-01	0.	327	.192442E-01	0.
296	.241096E-01	0.	312	.129790E-01	0.	328	.203023E-01	0.
297	.227290E-01	0.	313	.129144E-01	0.	329	.214619E-01	0.
298	.214621E-01	0.	314	.129144E-01	0.	330	.227288E-01	0.
299	.203025E-01	0.	315	.129789E-01	0.	331	.241094E-01	0.
300	.192443E-01	0.	316	.131084E-01	0.	332	.256105E-01	0.
301	.182824E-01	0.	317	.133034E-01	0.	333	.272397E-01	0.
302	.174119E-01	0.	318	.135649E-01	0.	334	.290051E-01	0.
303	.166285E-01	0.	319	.138943E-01	0.	335	.309155E-01	0.
304	.159282E-01	0.	320	.142931E-01	0.	336	.329805E-01	0.
305	.153076E-01	0.	321	.147634E-01	0.	337	.352103E-01	0.

IN SUBGRAPH NO 23

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
338	.352103E-01	0.	354	.147622E-01	0.	370	.153034E-01	0.
339	.329803E-01	0.	355	.142918E-01	0.	371	.159238E-01	0.
340	.309153E-01	0.	356	.138928E-01	0.	372	.166237E-01	0.
341	.290049E-01	0.	357	.135634E-01	0.	373	.174068E-01	0.
342	.272394E-01	0.	358	.133017E-01	0.	374	.182769E-01	0.
343	.256102E-01	0.	359	.131066E-01	0.	375	.192384E-01	0.
344	.241090E-01	0.	360	.129770E-01	0.	376	.202961E-01	0.
345	.227284E-01	0.	361	.129122E-01	0.	377	.214553E-01	0.
346	.214614E-01	0.	362	.129121E-01	0.	378	.227217E-01	0.
347	.203017E-01	0.	363	.129765E-01	0.	379	.241018E-01	0.
348	.192435E-01	0.	364	.131058E-01	0.	380	.256023E-01	0.
349	.182815E-01	0.	365	.133006E-01	0.	381	.272309E-01	0.
350	.174110E-01	0.	366	.135619E-01	0.	382	.289956E-01	0.
351	.166274E-01	0.	367	.138910E-01	0.	383	.309053E-01	0.
352	.159271E-01	0.	368	.142896E-01	0.	384	.329696E-01	0.
353	.153064E-01	0.	369	.147596E-01	0.	385	.351987E-01	0.

IN SUBGRAPH NO 24

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
386	.351968E-01	0.	402	.147204E-01	0.	418	.151741E-01	0.
387	.329659E-01	0.	403	.142470E-01	0.	419	.157850E-01	0.
388	.308998E-01	0.	404	.138448E-01	0.	420	.164748E-01	0.
389	.289882E-01	0.	405	.135118E-01	0.	421	.172470E-01	0.
390	.272216E-01	0.	406	.132464E-01	0.	422	.181054E-01	0.
391	.255910E-01	0.	407	.130472E-01	0.	423	.190543E-01	0.
392	.240884E-01	0.	408	.129132E-01	0.	424	.200985E-01	0.
393	.227063E-01	0.	409	.128438E-01	0.	425	.212432E-01	0.
394	.214377E-01	0.	410	.128386E-01	0.	426	.224941E-01	0.
395	.202762E-01	0.	411	.128977E-01	0.	427	.238575E-01	0.
396	.192162E-01	0.	412	.130212E-01	0.	428	.253402E-01	0.
397	.182522E-01	0.	413	.132098E-01	0.	429	.269495E-01	0.
398	.173795E-01	0.	414	.134644E-01	0.	430	.286937E-01	0.
399	.165937E-01	0.	415	.137864E-01	0.	431	.305812E-01	0.
400	.158909E-01	0.	416	.141773E-01	0.	432	.326217E-01	0.
401	.152675E-01	0.	417	.146391E-01	0.	433	.348253E-01	0.

IN SUBGRAPH NO 25

VARIABLE	VALUE	VARIABLE	VALUE	VARIABLE	VALUE

	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
434	.347668E-01	0.	450	.133878E-01	0.	466	.110439E-01	0.
435	.325044E-01	0.	451	.128167E-01	0.	467	.113522E-01	0.
436	.304045E-01	0.	452	.123097E-01	0.	468	.117173E-01	0.
437	.284566E-01	0.	453	.118643E-01	0.	469	.121410E-01	0.
438	.266510E-01	0.	454	.114781E-01	0.	470	.126253E-01	0.
439	.249787E-01	0.	455	.111494E-01	0.	471	.131728E-01	0.
440	.234312E-01	0.	456	.108764E-01	0.	472	.137862E-01	0.
441	.220069E-01	0.	457	.106578E-01	0.	473	.144685E-01	0.
442	.206836E-01	0.	458	.104925E-01	0.	474	.152231E-01	0.
443	.194638E-01	0.	459	.103797E-01	0.	475	.160538E-01	0.
444	.183442E-01	0.	460	.103187E-01	0.	476	.169649E-01	0.
445	.173164E-01	0.	461	.103094E-01	0.	477	.179607E-01	0.
446	.163751E-01	0.	462	.103516E-01	0.	478	.190463E-01	0.
447	.155157E-01	0.	463	.104455E-01	0.	479	.202272E-01	0.
448	.147339E-01	0.	464	.105917E-01	0.	480	.215092E-01	0.
449	.140258E-01	0.	465	.107908E-01	0.	481	.228988E-01	0.

IN SUBGRAPH NO 26

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
482	.260289E-01	0.	498	.777144E-01	0.	514	.239897E+00	0.
483	.277851E-01	0.	499	.833588E-01	0.	515	.257455E+00	0.
484	.296802E-01	0.	500	.894200E-01	0.	516	.276301E+00	0.
485	.317238E-01	0.	501	.959283E-01	0.	517	.296527E+00	0.
486	.339259E-01	0.	502	.102916E+00	0.	518	.318237E+00	0.
487	.362977E-01	0.	503	.110419E+00	0.	519	.341537E+00	0.
488	.388510E-01	0.	504	.118473E+00	0.	520	.366546E+00	0.
489	.415985E-01	0.	505	.127120E+00	0.	521	.393387E+00	0.
490	.445540E-01	0.	506	.136403E+00	0.	522	.422195E+00	0.
491	.477323E-01	0.	507	.146367E+00	0.	523	.453113E+00	0.
492	.511492E-01	0.	508	.157064E+00	0.	524	.486298E+00	0.
493	.548219E-01	0.	509	.168546E+00	0.	525	.521914E+00	0.
494	.587687E-01	0.	510	.180870E+00	0.	526	.560139E+00	0.
495	.630094E-01	0.	511	.194099E+00	0.	527	.601166E+00	0.
496	.675651E-01	0.	512	.208298E+00	0.	528	.645198E+00	0.
497	.724586E-01	0.	513	.223539E+00	0.	529	.692456E+00	0.

IN SUBGRAPH NO 27

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
530	.692612E+00	0.	546	.226835E+00	0.	562	.837522E-01	0.

531	.645511E+00	0.	547	.211936E+00	0.	563	.797011E-01	0.
532	.601638E+00	0.	548	.198048E+00	0.	564	.760486E-01	0.
533	.560773E+00	0.	549	.185149E+00	0.	565	.727763E-01	0.
534	.522712E+00	0.	550	.173176E+00	0.	566	.698679E-01	0.
535	.487265E+00	0.	551	.162069E+00	0.	567	.673089E-01	0.
536	.454253E+00	0.	552	.151772E+00	0.	568	.650864E-01	0.
537	.423513E+00	0.	553	.142234E+00	0.	569	.631893E-01	0.
538	.394891E+00	0.	554	.133407E+00	0.	570	.616081E-01	0.
539	.368243E+00	0.	555	.125248E+00	0.	571	.603350E-01	0.
540	.343436E+00	0.	556	.117714E+00	0.	572	.593636E-01	0.
541	.320347E+00	0.	557	.110770E+00	0.	573	.586890E-01	0.
542	.298859E+00	0.	558	.104379E+00	0.	574	.583079E-01	0.
543	.278865E+00	0.	559	.985094E-01	0.	575	.582182E-01	0.
544	.260266E+00	0.	560	.931329E-01	0.	576	.584197E-01	0.
545	.242968E+00	0.	561	.882220E-01	0.	577	.589133E-01	0.

IN SUBGRAPH NO 28

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
578	.557881E-01	0.	594	.214030E-01	0.	610	.174496E-01	0.
579	.521537E-01	0.	595	.204795E-01	0.	611	.179240E-01	0.
580	.487800E-01	0.	596	.196585E-01	0.	612	.184881E-01	0.
581	.456503E-01	0.	597	.189357E-01	0.	613	.191445E-01	0.
582	.427488E-01	0.	598	.183075E-01	0.	614	.198967E-01	0.
583	.400610E-01	0.	599	.177710E-01	0.	615	.207483E-01	0.
584	.375736E-01	0.	600	.173232E-01	0.	616	.217037E-01	0.
585	.352740E-01	0.	601	.169621E-01	0.	617	.227677E-01	0.
586	.331506E-01	0.	602	.166858E-01	0.	618	.239454E-01	0.
587	.311933E-01	0.	603	.164930E-01	0.	619	.252429E-01	0.
588	.293918E-01	0.	604	.163825E-01	0.	620	.266666E-01	0.
589	.277373E-01	0.	605	.163541E-01	0.	621	.282236E-01	0.
590	.262215E-01	0.	606	.164073E-01	0.	622	.299218E-01	0.
591	.248367E-01	0.	607	.165426E-01	0.	623	.317695E-01	0.
592	.235762E-01	0.	608	.167607E-01	0.	624	.337761E-01	0.
593	.224335E-01	0.	609	.170625E-01	0.	625	.359516E-01	0.

IN SUBGRAPH NO 29

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
626	.358536E-01	0.	642	.149677E-01	0.	658	.153635E-01	0.
627	.335797E-01	0.	643	.144830E-01	0.	659	.159787E-01	0.
628	.314736E-01	0.	644	.140706E-01	0.	660	.166739E-01	0.
629	.295250E-01	0.	645	.137287E-01	0.	661	.174524E-01	0.
630	.277239E-01	0.	646	.134553E-01	0.	662	.183181E-01	0.

631	.260615E-01	0.	647	.132493E-01	0.	663	.192755E-01	0.
632	.245293E-01	0.	648	.131095E-01	0.	664	.203293E-01	0.
633	.231198E-01	0.	649	.130352E-01	0.	665	.214846E-01	0.
634	.218259E-01	0.	650	.130261E-01	0.	666	.227475E-01	0.
635	.206412E-01	0.	651	.130822E-01	0.	667	.241240E-01	0.
636	.195596E-01	0.	652	.132037E-01	0.	668	.256212E-01	0.
637	.185759E-01	0.	653	.133911E-01	0.	669	.272464E-01	0.
638	.176850E-01	0.	654	.136456E-01	0.	670	.290080E-01	0.
639	.168825E-01	0.	655	.139682E-01	0.	671	.309145E-01	0.
640	.161645E-01	0.	656	.143607E-01	0.	672	.329756E-01	0.
641	.155273E-01	0.	657	.148250E-01	0.	673	.352016E-01	0.

IN SUBGRAPH NO 30

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
674	.351936E-01	0.	690	.146546E-01	0.	706	.149521E-01	0.
675	.329596E-01	0.	691	.141754E-01	0.	707	.155464E-01	0.
676	.308904E-01	0.	692	.137671E-01	0.	708	.162184E-01	0.
677	.289757E-01	0.	693	.134276E-01	0.	709	.169715E-01	0.
678	.272058E-01	0.	694	.131553E-01	0.	710	.178095E-01	0.
679	.255719E-01	0.	695	.129487E-01	0.	711	.187365E-01	0.
680	.240659E-01	0.	696	.128069E-01	0.	712	.197573E-01	0.
681	.226803E-01	0.	697	.127291E-01	0.	713	.208768E-01	0.
682	.214080E-01	0.	698	.127150E-01	0.	714	.221006E-01	0.
683	.202428E-01	0.	699	.127644E-01	0.	715	.234350E-01	0.
684	.191788E-01	0.	700	.128777E-01	0.	716	.248866E-01	0.
685	.182107E-01	0.	701	.130553E-01	0.	717	.264626E-01	0.
686	.173336E-01	0.	702	.132983E-01	0.	718	.281709E-01	0.
687	.165432E-01	0.	703	.136077E-01	0.	719	.300201E-01	0.
688	.158356E-01	0.	704	.139851E-01	0.	720	.320193E-01	0.
689	.152071E-01	0.	705	.144325E-01	0.	721	.341787E-01	0.

IN SUBGRAPH NO 31

VARIABLE	VALUE		VARIABLE	VALUE		VARIABLE	VALUE	
	REAL	IMAGINARY		REAL	IMAGINARY		REAL	IMAGINARY
722	.340217E-01	0.	739	.103365E-01	0.	756	.346725E-02	0.
723	.317047E-01	0.	740	.964783E-02	0.	757	.328655E-02	0.
724	.295461E-01	0.	741	.900737E-02	0.	758	.312227E-02	0.
725	.275352E-01	0.	742	.841195E-02	0.	759	.297361E-02	0.
726	.256621E-01	0.	743	.785859E-02	0.	760	.283982E-02	0.
727	.239172E-01	0.	744	.734452E-02	0.	761	.272022E-02	0.
728	.222920E-01	0.	745	.686718E-02	0.	762	.261423E-02	0.
729	.207732E-01	0.	746	.642417E-02	0.	763	.252131E-02	0.
730	.193682E-01	0.	747	.601328E-02	0.	764	.244099E-02	0.

731	.180552E-01	0.	748	.563246E-02	0.	765	.237288E-02	0.
732	.168324E-01	0.	749	.527980E-02	0.	766	.231663E-02	0.
733	.156937E-01	0.	750	.495353E-02	0.	767	.227197E-02	0.
734	.146335E-01	0.	751	.465204E-02	0.	768	.223867E-02	0.
735	.136466E-01	0.	752	.437381E-02	0.	769	.221656E-02	0.
736	.127278E-01	0.	753	.411744E-02	0.	770	.220553E-02	0.
737	.118727E-01	0.	754	.388167E-02	0.			
738	.110769E-01	0.	755	.366530E-02	0.			

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 785

CPU TIME: 1.423 SECONDS

```
EXAMPLE 16B

LADDER NETWORK WITH   770 NODES DECOMPOSED INTO    16 PROPER BLOCKS
NUMBER OF BASIC RECORDS    20
LENGTH OF ADDRESSING RECORDS    32
NUMBER OF SOLUTION RECORDS     8

MAXIMUM INTEGER AREA REQUIRED IS EQUAL TO 1063


MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO 6791


SOLUTION
--------

IN SUBGRAPH NO   1

 VARIABLE                VALUE

                  REAL          IMAGINARY

   785      .376037E-01   0.

IN SUBGRAPH NO   3

 VARIABLE                VALUE

                  REAL          IMAGINARY

   783      .597014E-01   0.

IN SUBGRAPH NO   6

 VARIABLE                VALUE

                  REAL          IMAGINARY

   789      .244028E-01   0.

IN SUBGRAPH NO  13

 VARIABLE                VALUE

                  REAL          IMAGINARY

   773      .743176E+00   0.
```

IN SUBGRAPH NO 26

VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY	VARIABLE	VALUE REAL	VALUE IMAGINARY
482	.260289E-01	0.	498	.777144E-01	0.	514	.239897E+00	0.
483	.277851E-01	0.	499	.833588E-01	0.	515	.257455E+00	0.
484	.296802E-01	0.	500	.894200E-01	0.	516	.276301E+00	0.
485	.317238E-01	0.	501	.959283E-01	0.	517	.296527E+00	0.
486	.339259E-01	0.	502	.102916E+00	0.	518	.318237E+00	0.
487	.362977E-01	0.	503	.110419E+00	0.	519	.341537E+00	0.
488	.388510E-01	0.	504	.118473E+00	0.	520	.366546E+00	0.
489	.415985E-01	0.	505	.127120E+00	0.	521	.393387E+00	0.
490	.445540E-01	0.	506	.136403E+00	0.	522	.422195E+00	0.
491	.477323E-01	0.	507	.146367E+00	0.	523	.453113E+00	0.
492	.511492E-01	0.	508	.157064E+00	0.	524	.486298E+00	0.
493	.548219E-01	0.	509	.168546E+00	0.	525	.521914E+00	0.
494	.587687E-01	0.	510	.180870E+00	0.	526	.560139E+00	0.
495	.630094E-01	0.	511	.194099E+00	0.	527	.601166E+00	0.
496	.675651E-01	0.	512	.208298E+00	0.	528	.645198E+00	0.
497	.724586E-01	0.	513	.223539E+00	0.	529	.692456E+00	0.

NUMBER OF VARIABLES IN THE SOLUTION VECTOR 52

CPU TIME: .290 SECONDS

Example 17

Ladder networks of different size were analyzed using data generated by the program LADDER listed on pp. 86-89. Figs. 16 and 17 show the dependence of the analysis time for the first execution versus number of network nodes. The larger the number of proper blocks the greater the time needed for mass storage operation. This is the reason that, for relatively small networks, the number of blocks should be small. From the tests performed for ladder networks, it can be seen that networks having less than 100 nodes need not be decomposed, but decomposition is necessary for networks having 1000 nodes or more. For large networks, both the computer memory required and time of analysis will be greater when the network is analyzed without decomposition. All subsequent analyses of the decomposed network will be much more efficient than those of the network analyzed without decomposition.

In Fig. 17 the curve a shows the dependence for partitioning into 4 proper blocks, curve b the case for 16 proper blocks and curve c the case for 128 proper blocks. In Fig. 18 curves b and c show the dependence for partitioning into 16 and 128 proper blocks, respectively, while curve a shows the estimated time for analysis without decomposition using the Harwell package [4-6].

Fig. 17 Computation time for different sized blocks (see text on page 146).

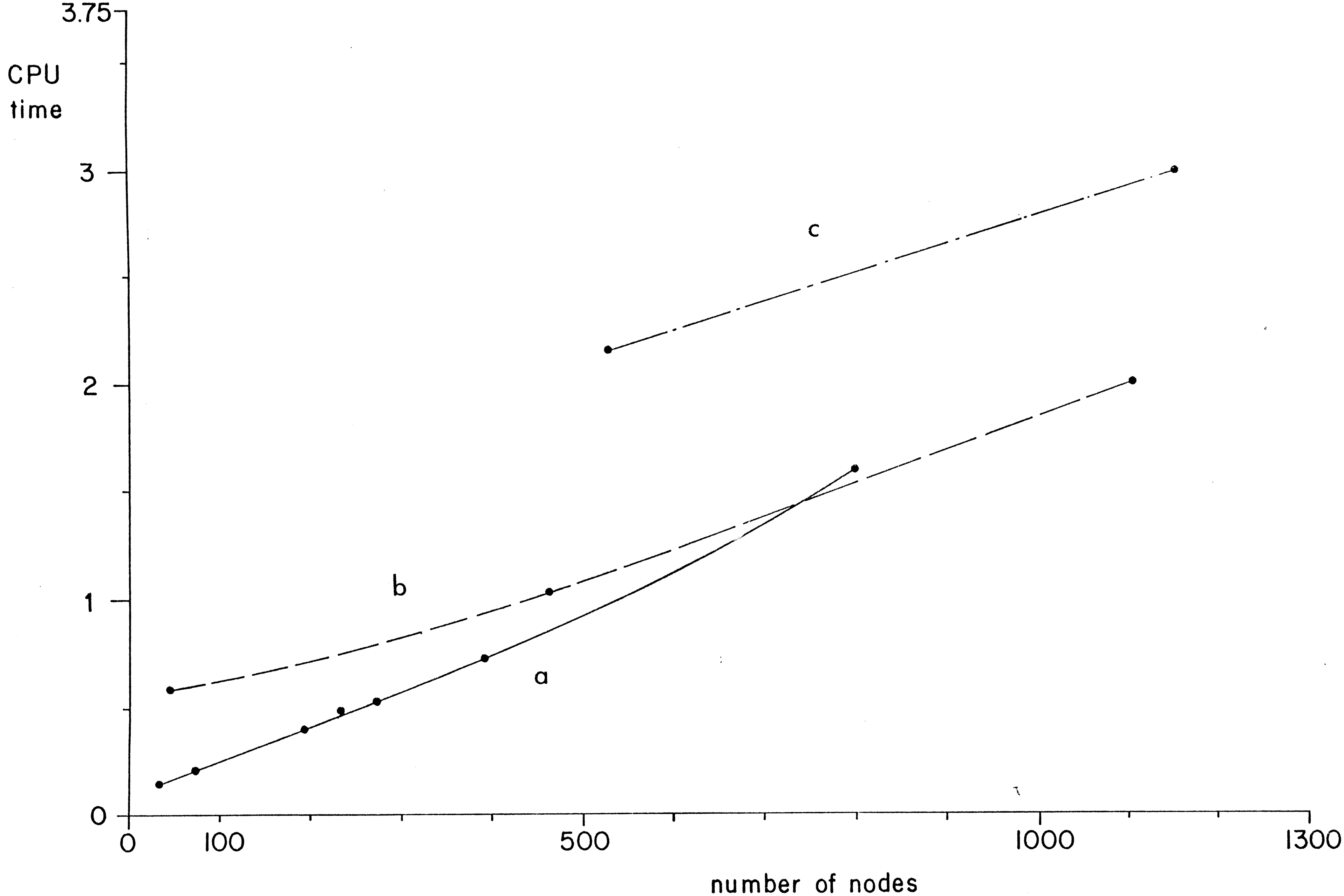
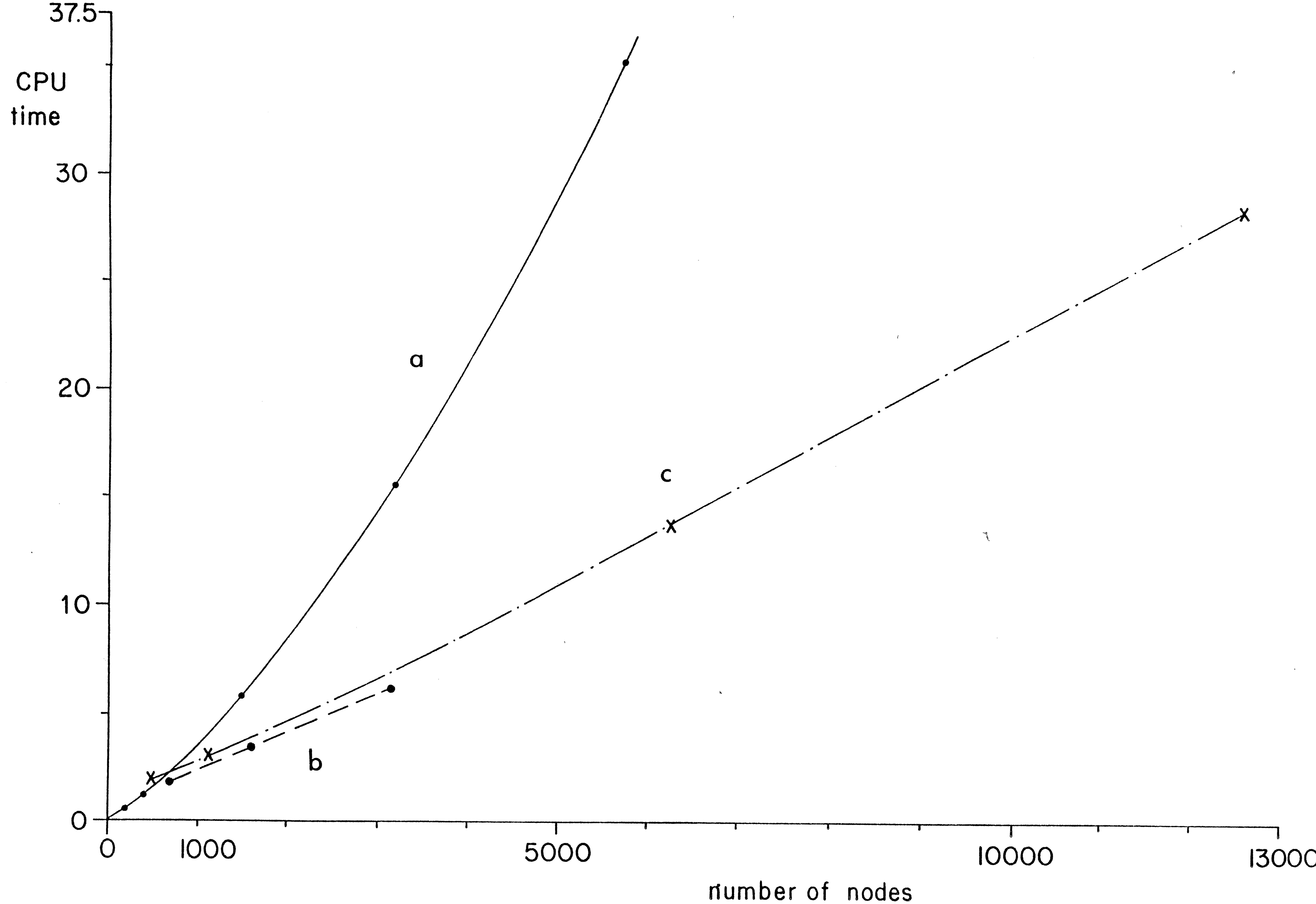


Fig. 18 Computation time for different approaches (see text on page 146).



REFERENCES

[1] W.K. Chen, Applied Graph Theory - Graphs and Electrical Networks. London: North-Holland, 1975.

[2] K. Zollenkopf, "Bifactorization - 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.

[3] "Document SOC-D20, CSSLE, sparse matrix techniques", Group on Simulation, Optimization and Control, Faculty of Engineering, McMaster University, Hamilton, Canada, September 1982.

[4] I.S. Duff, "MA28 - a set of Fortran subroutines for sparse unsymmetric linear equations", Computer Science and Systems Division, AERE Harwell, Oxfordshire, England, Report R.8730, November 1980.

[5] MA28A/B/C subroutine specification; Harwell Subroutine Library, AERE, Harwell, Oxfordshire, England, July 1980.

[6] "Document SOC-D1, MA28, sparse matrix techniques", Group on Simulation, Optimization and Control, Faculty of Engineering, McMaster University, Hamilton, Canada, December 1981. Origin of the package: Harwell Subroutine Library, AERE, Harwell, Oxfordshire, England.

APPENDIX

LISTING OF THE CSDSLE PACKAGE

Subroutine	Number of Lines (source text)	Number of Words (compiled code)	Listing from Page
CSDSLE1	357	1267	152
ASDNSR	216	1141	157
ABSUB	14	30	160
CHEBSUB	14	31	161
STOREF	10	41	161
READF	10	14	161
READSB	17	73	161
CHEDEL	22	101	162
ASUBSR	263	1566	162
LMIRHS	39	410	166
ANSUBN	145	1250	166
SMSSUB	73	122	169
CONNECT	48	374	170
EEDSUB	111	443	171
CONNUMB	87	130	172
ADDSUB	88	337	174
CASRHS	79	227	175
PSHDS	75	361	176
RMRHS	104	615	178
RSBR	78	340	179
RMERHS	54	140	180
RMIRHS	30	103	181
CASSOL	70	244	182
CSSLE1	178	1157	183
SSLECN	75	267	186
MSLECN	136	637	187
INVCN	118	416	189
MSLECC	73	365	191

Subroutine	Number of Lines (source text)	Number of Words (compiled code)	Listing from Page
MSLERC	28	127	192
PABIDES	33	245	192
DATAC	77	127	193
DATACN	29	72	194
DATBCN	47	155	195
SYMCN	97	231	195
SORDCN	129	276	197
REDUCN	76	232	199
SOLVCN	55	152	200
ASLECN	42	150	201
RSLECN	234	1444	202
FINDSYM	16	35	205
READL	16	41	206
CHNGE	49	132	206
PRINTIN	77	642	207
PRINTNC	42	314	208
PRINTRS	40	314	209
PRINT	41	316	209
PRINTI	48	330	210
PRINTS	60	372	211

```
      SUBROUTINE CSDSLE1(IN,INTWA,CMPLXWA,IM,SOLR)
C
C         THIS SUBROUTINE CALCULATES ADDRESSES FOR SUBMATRICES STORED
C
C         WITHIN WORKING AREAS INTWA AND CMPLXWA, AND CALLS THE
C
C         SUBROUTINES: ASDNSR WHICH ANALYSES THE SUBMATRICES OF THE
C
C         DECOMPOSED MATRIX, ANSUBSN WHICH ANALYSES THE SUBSTITUTE
C
C         SUBGRAPHS, AND PSHDS WHICH PERFORMS PARTIAL SOLUTION FOR
C
C         THE HIERARCHICALLY DECOMPOSED SYSTEM.
C
C         AUTHOR - JANUSZ STARZYK, MC MASTER UNIVERSITY, JANUARY 1983
C
      DIMENSION IN(1),INTWA(1),IM(1)
      COMPLEX CMPLXWA(1),SOLR(1)
C
C         IN THE MAIN PROGRAM THE USER MUST DETERMINE VALUES ONLY OF THE
C
C         FOLLOWING VARIABLES:  IN(4),IN(5),IN(7),IN(8),IN(9),IN(10),
C
C         IN(11),IN(12),IN(15),IN(16),IN(17),IN(18),IN(19),IN(20),
C
C         IN(22),IN(23),IN(25) UP TO THE LAST ELEMENT OF VECTOR IN,
C
C         IM(1),IM(2),IM(3),IM(4),IM(5).
C
C         IN(1)      NUMBER OF UNKNOWNS IN THE CURRENT SUBGRAPH
C
C         IN(2)      MAXIMUM AREA PREDICTED FOR THE SPARSE MATRIX
C                    DESCRIBING THE CURRENT SUBGRAPH
C
C         IN(3)      FLAG FOR INSUFFICIENT AREA
C
C         IN(4)      NUMBER OF SUBGRAPHS WHERE THE SOLUTION WILL
C                    BE RECALCULATED
C
C         IN(5)      NUMBER OF ALL PROPER BLOCKS
C
C         IN(6)      NUMBER OF INTERNAL NODES IN THE CURRENT SUBGRAPH
C
C         IN(7)       INDICATOR FOR PRINTING INTERMEDIATE RESULTS
C                 =0 ONLY PART OF INTERMEDIATE RESULTS WILL BE PRINTED
C                 =1 RESULTS WILL NOT BE PRINTED
C                 =2 ALL INTERMEDIATE RESULTS WILL BE PRINTED
C                 =3 ONLY THE SOLUTION WILL BE PRINTED
C
C         IN(8)       INDICATOR FOR KIND OF JOB
C                 =6 THE COMPLETE BIFACTORIZATION PROCESS IS
C                    EXECUTED FOR ALL PROPER BLOCKS
C                 =7 THE SOLUTION IS CALCULATED FOR AN ALTERED
C                    COEFFICIENT MATRIX AND/OR RIGHT-HAND SIDE VECTOR
C                    (PREVIOUS USE OF THE MAIN SUBROUTINE CSDSLE1 WITH
C                    IN(8)=6 IS REQUIRED)
C                 =8 THE SOLUTION IS CALCUTATED FOR AN ALTERED
C                    RIGHT-HAND SIDE VECTOR (PREVIOUS USE OF THE MAIN
C                    SUBROUTINE CSDSLE1 WITH IN(8)=6 IS REQUIRED)
C
C         IN(9)      THE HIGHEST INDEX OF THE SUBGRAPHS
C
C         IN(10)     NUMBER OF MODIFIED SUBGRAPHS, MUST BE LESS THAN
C                    IN(12)/2
C
```

```
C       IN(11)      LENGTH OF BASIC RECORD IN DOUBLE CM WORDS
C                   MUST BE GREATER THAN 256
C
C       IN(12)      LENGTH OF ADDRESSING RECORD IN CM WORDS
C
C       IN(13)      CURRENT NUMBER OF STORED BASIC RECORDS
C
C       IN(14)      CURRENT NUMBER OF SUBGRAPHS IN THE BASIC RECORD
C
C       IN(15)      MAXIMUM PREDICTED NUMBER OF BASIC AND SOLUTION
C                   RECORDS - TO STORE THE RESULTS OF ANALYSIS OF ALL
C                   SUBGRAPHS
C
C       IN(16)      MAXIMUM NUMBER OF EQUATIONS FOR ANY SUBGRAPH
C
C       IN(17)      MAXIMUM AREA PREDICTED FOR SPARSE MATRIX ANALYSIS
C                   OF SUBGRAPHS; MUST BE GREATER THAN IN(18)-IN(16)
C                   USUALLY 1-2 TIMES THE NUMBER OF NONZERO ELEMENTS
C
C       IN(18)      NUMBER OF NONZERO ELEMENTS IN THE MAXIMUM SUBMATRIX
C
C       IN(19)      THE LOWEST INDEX OVER ALL DECOMPOSITION NODES
C
C       IN(20)      NUMBER OF DATA RECORDS STORING INFORMATION
C                   ABOUT MODIFIED SUBGRAPHS
C
C       IN(21)      CURRENT NUMBER OF RHS RECORDS
C
C       IN(22)       INDICATOR FOR RHS UPDATING FORMULA
C                =0 COMPLETE SOLUTION VECTOR WILL BE RECALCULATED
C                =1 THE SOLUTION FOR MODIFIED SUBGRAPHS ONLY WILL BE
C                   RECALCULATED
C                =2 ONLY THE SOLUTION FOR THE SUBGRAPHS SPECIFIED BY THE
C                   USER WILL BE RECALCULATED - IF THE USER SPECIFIES
C                   SUBSTITUTE SUBGRAPHS ONLY THEN THE SOLUTION WILL
C                   BE RECALCUTATED AT THE PARTITION NODES ONLY
C
C       IN(23)       INDICATOR FOR OPERATIONS ON RANDOM FILES
C                =0 RANDOM ACCESS MULTI-RECORDS FILES WILL BE CREATED
C                =1 RANDOM ACCESS MULTI-RECORDS FILES WILL BE UPDATED
C                =2 THE PROGRAM WILL BE EXECUTED WITHOUT CREATING
C                   RANDOM ACCESS MULTI-RECORDS FILES
C                =3 RANDOM ACCESS MULTI-RECORDS FILES WILL BE EXTENDED
C
C       IN(24)      CURRENT NUMBER OF STORED ADDRESSING RECORDS
C
C       IN(25)      LENGTH OF DATA AND RHS RECORDS IN DOUBLE CM WORDS,
C                   MUST BE GREATER THAN 2*IN(10)
C
C       IN(26)      NUMBER OF ALL DECOMPOSITION NODES
C
C       IN(27)-IN(L)    STORE NUMBERS OF EXTERNAL NODES FOR THE
C                       SUBSTITUTE SUBGRAPHS, WHERE L=26+IN(9)/2
C
C       IN(L+1)-IN(M)   STORE THE INDICES OF ALL PROPER BLOCKS
C                       IN DECREASING ORDER, WHERE M=L+IN(5)
C
C       IN(M+1)-IN(K)   STORE THE INDICES OF SUBGRAPHS WHERE THE
C                       SOLUTION WILL BE RECALCULATED, WHERE K=M+IN(4)
C
C       IM(1)        MASS STORAGE INDICATOR
C                =0 MASS STORAGE FILES WILL NOT BE USED
C                =1 MASS STORAGE FILES WILL BE USED
C                =2 MASS STORAGE FILE NO. 2 (DATA FILE) WILL NOT
C                   BE USED
```

```
C                 =3 MASS STORAGE FILES WILL BE EXTENDED
C
C        IM(2)      FLAG DENOTING A REFERENCE TO THE MAIN SUBROUTINE
C                   MUST BE INITIALIZED BY THE USER TO 0 IN CONJUNCTION
C                   WITH THE FIRST CALL AND MUST NEVER BE SUBSEQUENTLY
C                   ALTERED
C
C        IM(3)      THE SIZE OF WORKING AREA INTWA
C
C        IM(4)      THE SIZE OF WORKING AREA CMPLXWA
C
C        IM(5)      NUMBER OF VARIABLES IN THE SOLUTION RECORD
C                   AT THE OUTPUT IM(5) CONTAINS THE NUMBER OF VARIABLES
C                   IN THE SOLUTION VECTOR
C
C        SOLR       SOLUTION RECORD
C                   FOR SMALL PROBLEMS IT CONTAINS SOLUTION VECTOR AND
C                   INDICES OF ITS VARIABLES
C                   IN SUCH CASE ITS SIZE MUST BE NOT LESS THAN THE
C                   DOUBLE THE NUMBER OF ALL VARIABLES
C
C        IF THE USER DOES NOT EMPLOY MASS STORAGE FILES FOR INPUT
C        DATA HE MUST PREPARE AND STORE THE DATA IN THE WORKING
C        AREA CMPLXWA STARTING FROM ELEMENT 4*IN(17)+1+3*IN(16)+IN(11)
C        THIS CORRESPONDS TO THE USE OF ONE DATA RECORD
C
C        THE SIZE OF THE WORKING AREA INTWA MUST NOT BE LESS THAN
C        3*IN(16)+2+2*IN(17)+2*IN(12)+2*IN(9)+IN(20)+1+IN(15)+1+
C        +(IN(9)-1)/IN(12)+3+IN(18)+IN(18)+7+IN(16)+IN(9)/2+
C        +(IN(26)/IN(25))*2+3+2*IN(9)+IN(15)+1+IN(9)/50+1=
C        =19+9*IN(9)/2+2*IN(12)+2*IN(15)+4*IN(16)+2*(IN(17)+IN(18))+
C        +IN(20)+(IN(9)-1)/IN(12)+IN(9)/50+(IN(26)/IN(25))*2
C
C        THE SIZE OF WORKING AREA CMPLXWA MUST NOT BE LESS THAN
C        IN(11)+3*IN(16)+4*IN(17)+IN(18)+IN(25)+1
C
C        FOR SMALL PROBLEMS OF UP TO 10 SUBGRAPHS THE USER IS ADVISED
C        TO SET STANDARD VALUES IN(9)=20,IN(12)=20,IN(15)=11,IN(20)=10,
C        AND USE INTWA OF SIZE 181+4*IN(16)+2(IN(17)+IN(18))
C
      CALL SECOND(TM1)
      MSI=IM(1)
      NRI=IM(2)
C
C        CHECK INPUT INFORMATION
C
      IF(IN(7).EQ.1.OR.IN(7).EQ.3) GOTO 3
      CALL PRINTIN(IN)
    3 CONTINUE
      IF(IN(9).GE.IN(5).AND.IN(5).GE.IN(10)) GOTO 1
      WRITE(6,100)
      RETURN
  100 FORMAT(1H ,/," THE NUMBER OF MODIFIED SUBGRAPHS SHOULD BE LESS THA
     +N OR EQUAL TO",/," THE NUMBER OF PROPER SUBGRAPHS AND THE HIGHEST
     +INDEX OF THE SUBGRAPHS",/," CHECK IN(5), IN(9) AND IN(10)",/)
    1 CONTINUE
      IF(IN(15).LE.IN(9)) GOTO 2
      WRITE(6,110)
  110 FORMAT(1H ,/," THE NUMBER OF BASIC RECORDS NEED NOT TO BE GREATER
     +THAN THE HIGHEST INDEX OF THE SUBGRAPHS",/," CHECK IN(15)",/)
      RETURN
    2 CONTINUE
      IF(IN(4).LE.IN(5).AND.(IN(5)*2-1).LE.IN(9)) GOTO 4
      WRITE(6,120)
  120 FORMAT(1H ,/," IN(4) SHOULD NOT BE GREATER THAN IN(5) AND (IN(5)*2
```

```
     +-1)<=IN(9)")
      RETURN
    4 IF(IN(27).EQ.0) GOTO 5
      WRITE(6,130)
  130 FORMAT(1H ,/," THE NUMBER OF EXTERNAL NODES FOR SUBSTITUTE SUBGRAP
     +H NO.1 MUST BE ZERO",/," CHECK IN(27)",/)
      RETURN
    5 CONTINUE
      IF(IN(8).GT.6.AND.MSI.EQ.0.AND.NRI.EQ.0) WRITE(6,140)
  140 FORMAT(1H ,/," WARNING!!! - DATA IS TAKEN TO BE IN BIFACTORIZED FO
     +RM")
      IF(IN(11).GT.256) GOTO 10
      WRITE(6,220)
  220 FORMAT(1H ,/," LENGTH OF BASIC RECORD MUST BE GREATER THAN 256"
     +,/)
   10 CONTINUE
      IF(IM(1).EQ.3) IN(23)=3
      IF(IN(23).EQ.3) IM(1)=3
      IF(IN(23).NE.1.AND.IN(23).NE.3.OR.IM(1).NE.0)GOTO 14
      WRITE(6,230)
  230 FORMAT(1H ,/," WITH IN(23) EQUAL TO 1 OR 3 MASS STORAGE FILES MUST
     + BE USED",/," CHECK IM(1)",/)
      RETURN
   14 CONTINUE
      IF(IN(17).LT.(IN(18)-IN(16)))WRITE(6,240)
  240 FORMAT(1H ,/," WARNING!!! - IN(17) MAY BE TOO SMALL",/)
      N=IN(16)
      LES=IN(17)
      LADR=IN(12)
      MNS=IN(9)
      IA1=3*N+2
      IA2=IA1+2*LES
      IA3=IA2+2*LADR
      IA4=IA3+2*MNS
      IA5=IA4+IN(20)+1
      IA6=IA5+IN(15)+1
      IA7=IA6+(IN(9)-1)/LADR+3
      IA8=IA7+IN(18)
      IA9=IA8+IN(18)
      IA10=IA9+7
      IA11=IA10+N
      IA12=IA11+MNS/2
      IA13=IA12+(IN(26)/IN(25)+1)*2+1
      IA14=IA13+2*IN(9)
      IA15=IA14+IN(15)+1
C
C         CHECK THE SIZE OF WORKING AREA INTWA
C
      IA16=IA15+IN(9)/50+1
      IF(IM(3).LT.IA16) GOTO 6
      IF(IM(3).GT.IA16) WRITE(6,180)IA16
      IB1=4*LES+1
      IB2=IB1+2*N
      IB3=IB2+N
      IB4=IB3+IN(11)
      IB5=IB4+IN(25)
C
C         CHECK THE SIZE OF WORKING AREA CMPLXWA
C
      IB6=IB5+IN(18)
      IF(IM(4).LT.IB6) GOTO 7
      IF(IM(4).GT.IB6) WRITE(6,170) IB6
C
C         THE ADDRESSES CALCULATED IN THIS SUBROUTINE ARE NEEDED FOR
C
```

```
C         PARTINIONING THE WORKING AREAS INTO ARRAYS TO BECOME THE
C
C         PARAMETERS OF SUBROUTINE ASDNSR. IN SUBROUTINE ASDNSR THE
C
C         ARRAYS HAVE THE FOLLOWING DIMENSIONS
C
C         IN(26+IN(9)/2+IN(5)+IN(4)),
C
C         LCOL(3*N+1),ITAG(2*LES),CE(4*LES),DE(2*N),V(N),IADREC(2*LADR)
C
C         BREC(LREC),DREC(IN(25)),NSUB(2*MNS),INDEX2(IN(20)+1)
C
C         INDEX3(IN(15)+1),INDEX5((MNS-1)/LADR+3),AK(NEL),NCOL(NEL),
C
C         NROW(NEL),INDX(7),NON(N),INDEX1(IN(26)/IN(25)+2),NASUB(MNS*2)
C
C         INDEX7(IN(15)+1),IT(IN(9)/50+1)
C
C         WHERE MNS=IN(9),NMS=IN(10),LREC=IN(11),LADR=IN(12),N=IN(16)
C
C         LES=IN(17),NEL=IN(18)
C
C         SUBROUTINE ASDNSR PERFORMS AN ANALYSIS OF EACH SUBGRAPH OF
C
C         THE DECOMPOSED GRAPH AND STORES THE RESULTS
C
      CALL ASDNSR(IN,INTWA(1),INTWA(IA1),CMPLXWA(1),CMPLXWA(IB1)
     +,CMPLXWA(IB2),INTWA(IA2),CMPLXWA(IB3),CMPLXWA(IB4),INTWA(IA3),
     +INTWA(IA4),INTWA(IA5),INTWA(IA6),CMPLXWA(IB5),INTWA(IA7),
     +INTWA(IA8),INTWA(IA9),INTWA(IA10),INTWA(IA12),INTWA(IA13),
     +INTWA(IA14),NRI,MSI,INTWA(IA15),IA6,IA16)
      IF(IN(3).EQ.1) RETURN
      IM(2)=NRI
C
C         IN ANSUBSN WE HAVE AN ADDITIONAL MATRIX
C
C         NEXTN(MNS/2)
C
C         SUBROUTINE ANSUBSN PERFORMS ANALYSIS OF SUBSTITUTE GRAPHS
C
      CALL ANSUBSN(IN,INTWA(1),INTWA(IA1),CMPLXWA(1),CMPLXWA(IB1)
     +,CMPLXWA(IB2),INTWA(IA2),CMPLXWA(IB3),INTWA(IA3),
     +INTWA(IA5),INTWA(IA6),CMPLXWA(IB5),INTWA(IA7),
     +INTWA(IA8),INTWA(IA9),INTWA(IA10),INTWA(IA11),
     +CMPLXWA(IB4),INTWA(IA12),INTWA(IA15))
      IF(IN(3).EQ.1)RETURN
C
C         CHECK WHETHER ALL SUBSTITUTE SUBGRAPHS HAVE BEEN ANALYZED
C
      CALL CHEBSUB(1,INTWA(IA15),IR)
      IF(IR.EQ.1)GOTO 12
      WRITE(6,210)
      GOTO 13
  210 FORMAT(1H ,/," HIERARCHICAL STRUCTURE IS NOT COMPLETE",/," SOLUTIO
     +N CANNOT BE CALCULATED",/)
C
C         SUBROUTINE PSHDS PERFORMS A PARTIAL SOLUTION FOR THE
C
C         HIERARCHICALLY DECOMPOSED SYSTEM
C
   12 CALL PSHDS(IN,INTWA(1),INTWA(IA1),CMPLXWA(1),CMPLXWA(IB1)
     +,CMPLXWA(IB2),INTWA(IA2),CMPLXWA(IB3),INTWA(IA3),
     +INTWA(IA5),INTWA(IA6),INTWA(IA9),INTWA(IA10),INTWA(IA12),
     +INTWA(IA14),CMPLXWA(IB4),SOLR,IM)
      IF(IN(3).EQ.1) RETURN
```

```
      CALL SECOND(TM2)
      CPU=TM2-TM1
      IF(IN(7).NE.1)WRITE(6,190)CPU
  190 FORMAT(1H ,/," CPU TIME:",F8.3," SECONDS",/)
      IF(IN(23).EQ.2)RETURN
C
C          STORE MASS STORAGE FILES INFORMATION
C
   13 LI=26+IN(9)/2+IN(5)+IN(4)
      IF(LI.GE.IA4)GOTO 9
      DO 8 I=1,LI
    8 INTWA(I)=IN(I)
      IND=(IN(9)-1)/IN(12)+2
      CALL STOREF(5,INTWA,IA16,IND,INTWA(IA6))
      RETURN
    9 WRITE(6,200)
  200 FORMAT(1H ,/," MASS STORAGE INFORMATION CANNOT BE STORED",/," INCR
     +EASE IN(12),IN(16) OR IN(17)",/)
      RETURN
    6 WRITE(6,150)
      IN(3)=3
      RETURN
    7 WRITE(6,160)
      IN(3)=4
      RETURN
  150 FORMAT(1H ,/," INTEGER WORKING AREA INTWA TOO SMALL - CHECK IM(3)"
     +,/)
  160 FORMAT(1H ,/," COMPLEX WORKING AREA CMPLXWA TOO SMALL - CHECK IM(4
     +)",/)
  170 FORMAT(1H ,/," MAXIMUM COMPLEX AREA REQUIRED IS EQUAL TO",I5,/)
  180 FORMAT(1H ,/," MAXIMUM INTEGER AREA REQUIRED IS EQUAL TO",I5,/)
      END
C
C
      SUBROUTINE ASDNSR(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,DREC,NSUB,INDEX
     +2,INDEX3,INDEX5,AK,NCOL,NROW,INDX,NON,INDEX1,NASUB,INDEX7,NRI,MSI,
     +IT,IA6,IA16)
C
C          THIS SUBROUTINE PERFORMS ANALYSIS OF SUBGRAPHS OF A
C
C          DECOMPOSED GRAPH AND STORES THE RESULTS IN RANDOM FILE
C
C          NO.3, WHILE ADDRESSES FOR THESE RESULTS ARE STORED IN
C
C          RANDOM FILE NO.5. SUBGRAPHS TO BE ANALYZED HAVE TO BE
C
C          STORED IN RANDOM FILE NO.2 OR IN THE WORKING AREA CMPLXWA
C
      DIMENSION IN(1),LCOL(1),ITAG(1),IADREC(1),NSUB(1),INDEX1(1)
     +,INDEX2(1),INDEX3(1),INDEX5(1),NCOL(1),NROW(1),INDX(1),NON(1)
     +,NASUB(1),INDEX7(1),IT(1)
      COMPLEX CE(1),DE(1),V(1),BREC(1),AK(1),DREC(1)
C
C          LTHR             LENGTH OF SPACE FOR CURRENT BLOCK IN THE BASIC
C                           RECORD
C
C          BREC             BASIC RECORD
C
C          IADREC           ADDRESSING RECORD
C
C          NON              ORIGINAL INDICES OF THE NODES
C
C          CURRENT INFORMATION ABOUT RANDOM FILES IS STORED IN THE ARRAY
C
C          INDX, WHERE
```

```
C
C        INDX(2)     :    CURRENT INDEX OF DATA RECORD IN RANDOM FILE
C                         NO. 2
C
C        INDX(3)     :    CURRENT INDEX OF A BASIC RECORD IN RANDOM FILE
C                         NO. 3
C
C        INDX(4)     :    CURRENT INDEX OF THE RHS RECORD IN
C                         RANDOM FILE NO. 1
C
C        INDX(5)     :    CURRENT INDEX OF THE ADDRESSING RECORD IN
C                         RANDOM FILE NO. 5
C
C        INDX(7)     :    CURRENT INDEX OF THE SOLUTION RECORD IN
C                         RANDOM FILE NO. 7
C
      MNS=IN(9)
      LREC=IN(11)
      LADR=IN(12)
      LDREC=2*LREC
      LLDR=LADR*2
      NMS=IN(10)
      IF(IN(23).EQ.3) GOTO 10
      IAX=IN(9)/50+1
      DO 11 I=1,IAX
   11 IT(I)=0
      BREC(1)=CMPLX(514.,0.)
      INDX(2)=1
      INDX(3)=1
      INDX(4)=1
      INDX(5)=1
      INDX(7)=1
      IN(13)=0
      IN(14)=0
      IN(21)=1
      IN(24)=0
   10 MMNS=2*MNS
      IF(NRI.GT.0) GOTO 21
      NRI=1
      IF(MSI.EQ.0) GOTO 7
C
C        OPEN RANDOM FILES
C
      IREC1=(IN(26)/IN(25)+1)*2+1
      IREC2=IN(20)+1
      IREC3=IN(15)+1
      IREC5=(MNS-1)/LADR+3
      CALL OPENMS(5,INDEX5,IREC5,0)
C
C        READ STORED MASS STORAGE FILES INFORMATION
C
      IF(IN(23).NE.1.AND.IN(23).NE.3)GOTO 15
      IND=(IN(9)-1)/IN(12)+2
      CALL READF(5,LCOL,IA16,IND,LCOL(IA6))
      IN(5)=LCOL(5)
      IN(9)=LCOL(9)
      IN(13)=INDX(3)-1
      IN(14)=LCOL(14)
      IN(19)=LCOL(19)
      IN(21)=LCOL(21)
      IN(24)=LCOL(24)
      IN(26)=LCOL(26)
      IF(MSI.EQ.2)INDX(2)=1
   15 CONTINUE
      CALL OPENMS(1,INDEX1,IREC1,0)
```

```
      IF(MSI.NE.2) CALL OPENMS(2,INDEX2,IREC2,0)
      CALL OPENMS(3,INDEX3,IREC3,0)
      CALL OPENMS(7,INDEX7,IREC3,0)
   21 IF(IN(23).EQ.0.OR.IN(23).EQ.2) GOTO 5
C
C          READ THE INFORMATION STORED IN THE RANDOM FILES
C
      CALL READMS(5,IADREC,LLDR,INDX(5))
      CALL READMS(3,BREC,LDREC,INDX(3))
    5 IF(MSI.NE.1.AND.MSI.NE.3) GOTO 7
C
C          READ THE FIRST DATA RECORD FROM RANDOM FILE NO.2.
C
C          INDICES OF MODIFIED SUBGRAPHS ARE STORED IN THE FIRST NMS
C
C          ELEMENTS OF MATRIX NSUB STORED IN THE FIRST RECORD OF
C
C          RANDOM FILE NO.2 AT THE BEGINNING OF THE INFORMATION ON
C
C          SUBGRAPHS. ELEMENT NMS+K, 1<=K<=NMS OF MATRIX NSUB STORES THE
C
C          LENGTH OF THE RECORD ASSOCIATED WITH THE KTH SUBGRAPH
C
      LDDX=IN(25)*2
      CALL READMS(2,DREC,LDDX,1)
    7 CONTINUE
      NSBR=IN(14)
      NMS2=2*NMS
      DO 6 I=1,NMS2
      NSUB(I)=INT(REAL(DREC(I)))
    6 NASUB(I)=INT(AIMAG(DREC(I)))
      DO 14 I=1,NMS
   14 CALL ABSUB(NSUB(I),IT)
      IF(IN(7).EQ.1.OR.IN(7).EQ.3) GOTO 1
      WRITE(6,123)(NSUB(I),I=1,NMS)
    1 CONTINUE
123   FORMAT(1H ,/," INDICES OF MODIFIED SUBGRAPHS",//,(20I4))
      DO 2 NBS=1,NMS
C
C          READ THE MODIFIED SUBGRAPH HAVING INDEX NSUB(NBS)
C
      IF(IN(8).LT.8) CALL READSB(CE,NSUB,NASUB,NBS,DREC,INDX,LDDX,NMS)
      IF(IN(8).EQ.8) CALL READSB(V,NSUB,NASUB,NBS,DREC,INDX,LDDX,NMS)
      IF(IN(8).EQ.8) GOTO 8
C
C          FOR AN UPDATED COEFFICIENT MATRIX AND RHS VECTOR
C
C          INFORMATION ABOUT THE SUBGRAPH IS STORED IN THE REAL AND
C
C          IMAGINARY PARTS OF COMPLEX MATRIX CE IN THE SEQUENCE
C
C          (N,NINT),(LES,NEL),AK,(NROW,NCOL),V,NON
C
C          WHEN ONLY THE RHS IS UPDATED THE VECTOR V CONTAINS THIS RHS
C
      IN(1)=INT(REAL(CE(1)))
      NEL=INT(AIMAG(CE(2)))
      N=IN(1)
      IN(2)=INT(REAL(CE(2)))
      IN(3)=2
      IN(6)=INT(AIMAG(CE(1)))
C
C          CHECK INPUT DATA DESCRIBING NBS-TH SUBNETWORK
C
      IF(N.GT.IN(16)) GOTO 12
```

```
      IF(IN(6).GT.N) GOTO 16
      IF(IN(2).GT.IN(17)) GOTO 17
      IF(NEL.GT.IN(18))GOTO 18
      DO 3 I=1,NEL
      AK(I)=CE(I+2)
      NROW(I)=INT(REAL(CE(I+NEL+2)))
    3 NCOL(I)=INT(AIMAG(CE(I+NEL+2)))
      IX=2*NEL+2
      DO4 I=1,N
    4 V(I)=CE(I+IX)
      IF(IN(8).EQ.7) GOTO 8
      IX=IX+N
      IY=(N-1)/2
      DO 13 I=1,IY
      NON(2*I)=INT(AIMAG(CE(IX+I)))
   13 NON(2*I-1)=INT(REAL(CE(IX+I)))
      IY=IY+1
      NON(2*IY-1)=INT(REAL(CE(IX+IY)))
      IF((2*IY).EQ.N) NON(2*IY)=INT(AIMAG(CE(IX+IY)))
C
C          CHECK DIAGONAL ELEMENTS OF THE SUBMATRIX
C
      CALL CHEDEL(IN(6),NEL,NCOL,NROW,NSUB,NBS,LCOL,IN(3))
      IF(IN(3).EQ.1)RETURN
C
C          ANALYSE THE SUBGRAPH AND STORE THE RESULTS IN THE BASIC
C          RECORD
C
    8 IN3=2
      CALL ASUBSR(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,NSUB,INDEX3,
     +INDEX5,AK,NCOL,NROW,INDX,NON,N,NEL,NBS,NSBR,LREC,LADR,IN3)
      IF(IN(3).EQ.1) RETURN
    2 CONTINUE
      RETURN
   12 WRITE(6,130)NSUB(NBS)
      IN(3)=1
      RETURN
  130 FORMAT(1H ,/," NUMBER OF UNKNOWNS IN THE SUBGRAPH",I4," IS GREATER
     + THAN IN(16)",/)
   16 WRITE(6,140)NSUB(NBS)
      IN(3)=1
      RETURN
  140 FORMAT(1H ,/," NUMBER OF INTERNAL NODES IN THE SUBGRAPH",I4," IS G
     +REATER THAN THE NUMBER OF UNKNOWNS IN THIS SUBGRAPH",/)
   17 WRITE(6,150)NSUB(NBS)
      IN(3)=1
      RETURN
  150 FORMAT(1H ,/," LENGTH OF THE AREA PREDICTED FOR SPARSE MATRIX OPER
     +ATIONS IN THE SUBGRAPH",I4," IS GREATER THAN IN(17)",/)
   18 WRITE(6,160)NSUB(NBS)
      IN(3)=1
      RETURN
  160 FORMAT(1H ,/," NUMBER OF NONZERO ELEMENTS IN THE SUBGRAPH",I4," IS
     + GREATER THAN IN(18)",/)
      END
C
C
      SUBROUTINE ABSUB(N,IT)
C
C          THIS SUBROUTINE ADDS BINARY CODE OF A SUBNETWORK NO. N TO THE
C
C          VECTOR IT
C
      DIMENSION IT(1)
      IB=1
```

```
      IA=(N-1)/50+1
      IC=N-(IA-1)*50-1
      ID=SHIFT(IB,IC)
      IT(IA)=OR(IT(IA),ID)
      RETURN
      END
C
C
      SUBROUTINE CHEBSUB(N,IT,IR)
C
C         THIS SUBROUTINE CHECKS BINARY CODE OF A SUBNETWORK NO. N
C
C         IN THE VECTOR IT
C
      DIMENSION IT(1)
      IB=1
      IA=(N-1)/50+1
      IC=N-(IA-1)*50-1
      ID=SHIFT(IB,IC)
      IR=AND(IT(IA),ID)
      RETURN
      END
C
C
      SUBROUTINE STOREF(NR,INTWA,LGTH,IND,INDEX5)
C
C         THIS SUBROUTINE STORES INTEGER VECTOR TO MASS STORAGE FILE
C
C         NO. NR
C
      INTEGER INTWA(1),INDEX5(1)
      CALL WRITMS(NR,INTWA,LGTH,IND)
      RETURN
      END
C
C
      SUBROUTINE READF(NR,INTWA,LGTH,IND,INDEX5)
C
C         THIS SUBROUTINE READS INTEGER VECTOR FROM MASS STORAGE FILE
C
C         NO. NR
C
      INTEGER INTWA(1),INDEX5(1)
      CALL READMS(NR,INTWA,LGTH,IND)
      RETURN
      END
C
C
      SUBROUTINE READSB(CE,NSUB,NASUB,NBS,DREC,INDX,LDREC,NMS)
C
C         THIS SUBROUTINE READS INPUT DATA FOR THE SUBGRAPH
C
C         WHOSE INDEX IS NBS
C
      INTEGER NSUB(1),NASUB(1),INDX(1)
      COMPLEX CE(1),DREC(1)
      IF(NASUB(NBS).EQ.INDX(2)) GOTO 1
      CALL READMS(2,DREC,LDREC,NASUB(NBS))
      INDX(2)=NASUB(NBS)
    1 IAD=NASUB(NMS+NBS)-1
      IG=NSUB(NMS+NBS)
      DO 2 I=1,IG
    2 CE(I)=DREC(IAD+I)
      RETURN
      END
```

```
C
C
      SUBROUTINE CHEDEL(NINT,NEL,NCOL,NROW,NSUB,NBS,L,NN)
C
C         THIS SUBROUTINE CHECKS WHETHER ALL DIAGONAL ELEMENTS ARE
C
C         DEFINED IN THE SUBGRAPH NSUB(NBS)
C
      INTEGER NCOL(1),NROW(1),NSUB(1),L(1)
      NN=0
      DO 1 I=1,NINT
    1 L(I)=0
      DO 2 I=1,NEL
    2 IF(NCOL(I).EQ.NROW(I)) L(NCOL(I))=1
      IL=0
      DO 3 I=1,NINT
    3 IL=IL+L(I)
      IF(IL.EQ.NINT)RETURN
      WRITE(6,100)NSUB(NBS)
  100 FORMAT(1H ,/," ALL DIAGONAL ELEMENTS SHOULD BE DEFINED IN THE SUBM
     +ATRIX NO.",I5)
      NN=1
      RETURN
      END
C
C
      SUBROUTINE ASUBSR(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,NSUB,INDEX3,
     +INDEX5,AK,NCOL,NROW,INDX,NON,N,NEL,NBS,NSBR,LREC,LADR,IN3)
C
C         THIS SUBROUTINE ANALYSES THE SUBGRAPH AND STORES THE RESULTS
C
C         OF ANALYSIS IN THE BASIC RECORD BREC
C
C         THE REAL PARTS OF THE FIRST 256 ELEMENTS OF EACH BASIC RECORD
C
C         CONTAIN THE ADDRESSES OF THE RESULTS FOR DIFFERENT SUBGRAPHS.
C
C         RESULTS ARE STORED IN THE COMPLEX VECTOR BREC IN THE SEQUENCE:
C
C         (N,NINT),(LES,LTHR),(LCOL(1),LCOL(N+2)),(LCOL(N+1),LCOL(2N+2))
C
C         (ITAG(1),(ITAG(LES+1)),CE,DE,V
C
      DIMENSION IN(1),LCOL(1),ITAG(1),IADREC(1),NSUB(1),INDEX3(1),
     +INDEX5(1),NCOL(1),NROW(1),INDX(1),NON(1)
      COMPLEX CE(1),DE(1),V(1),BREC(1),AK(1),COEFF
C
      LLDR=LADR*2
      LDREC=2*LREC
      LES=IN(2)
      LTHR=(2+5*N+5*LES)*2
      NMS=IN(10)
C
C         PRINT THE COEFFICIENT MATRIX
C
      IF(IN(7).EQ.1.OR.IN(7).EQ.3) GOTO 15
      IF(IN(8).EQ.8) GOTO 28
      WRITE(6,20) NSUB(NBS)
      CALL PRINTNC(NEL,NROW,NCOL,AK)
      WRITE(6,49)
      CALL PRINT(N,V)
49    FORMAT(1H ,/," RHS VECTOR ",/)
15    CONTINUE
      IF(IN(8).EQ.7) GOTO 21
      IF(IN(8).EQ.8) GOTO 19
```

```
C
C        THIS PART IS EXECUTED WHEN THE SUBGRAPH IS ANALYSED FOR THE
C        FIRST TIME
C
      CALL DATAC(N,NEL,AK,NCOL,NROW,LCOL,ITAG,CE,DE,IN(2),IN(3))
      IF(IN(3).EQ.1) RETURN
   20 FORMAT(1H ,/," COEFFICIENT MATRIX OF SUBGRAPH NO",I3,/)
      IN(3)=2
      IF(IN3.EQ.0) IN(3)=0
C
C        PERFORM PARTIAL BIFACTORIZATION OF THE SUBGRAPH MATRIX
C
      CALL CSSLE1(IN,LCOL,ITAG,CE,DE,V,EPS)
      IF(IN(3).EQ.1) RETURN
      LES=IN(2)
      NINT=IN(6)
C
C        MULTIPLY LEFT PART OF LR FACTORIZATION BY THE RHS VECTOR FOR
C        THE NBS SUBGRAPH
C
      CALL LMIRHS(NINT,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),CE,DE,V)
C
      LTHR=(2+5*N+5*LES)*2
      IF(IN(23).NE.1) GOTO 1
C
C        READ THE ADDRESS FOR THE SUBGRAPH NSUB(NBS) AND THE INDEX
C        OF THE BASIC RECORD
C
      NADR=(NSUB(NBS)-1)/LADR+1
      IADR=NSUB(NBS)-(NADR-1)*LADR
      IF(INDX(5).EQ.NADR) GOTO 2
      CALL WRITMS(5,IADREC,LLDR,INDX(5),1,1)
      CALL READMS(5,IADREC,LLDR,NADR)
      INDX(5)=NADR
C
C        READ BASIC RECORD
C
    2 NRBR=IADREC(IADR)
      NSBR=IADREC(IADR+LADR)
      IF(INDX(3).EQ.NRBR) GOTO 3
      IF(INDX(3).GT.IN(15)) GOTO 14
      CALL WRITMS(3,BREC,LDREC,INDX(3),1,1)
      CALL READMS(3,BREC,LDREC,NRBR)
      INDX(3)=NRBR
      GOTO 3
    1 NSBR=NSBR+1
      IN(14)=IN(14)+1
      IF(INDX(3).GT.IN(13)) GOTO 17
      INDX(3)=IN(13)+1
      CALL READMS(3,BREC,LDREC,INDX(3))
C
C        IF NSBR>255 OPEN NEW BASIC RECORD
C
   17 IF(NSBR.GT.255) GOTO 13
    3 IADR1=INT(REAL(BREC(NSBR)))
      IADR=IADR1+LTHR
C
C        IF IADR>2*LREC THEN STORE THE BASIC RECORD AND OPEN A NEW ONE.
C
C        BASIC RECORDS ARE STORED IN THE RANDOM FILE NO.3, EACH HAVING
C
C        LREC COMPLEX ELEMENTS
C
      IF(IADR.LE.LDREC)GOTO 10
   13 IN(13)=IN(13)+1
```

```
      INDX(3)=IN(13)+1
      IF(IN(23).NE.1) GOTO.4
      WRITE(6,130)NSUB(NBS)
  130 FORMAT(1H ,/," MODIFIED SUBGRAPH NO.",I4," REQUIRES MORE STORAGE T
     +HAN THE PREVIOUS ONE",/," REPEAT ANALYSIS FOR ALL SUBGRAPHS",/)
      IN(3)=1
      RETURN
    4 IF(IN(13).GT.IN(15)) GOTO 14
      CALL WRITMS(3,BREC,LDREC,IN(13))
      DO 9 I=2,256
    9 BREC(I)=CMPLX(0.,0.)
      NSBR=1
      IN(14)=1
      IADR1=514
      IADR=514+LTHR
      IF(IADR.GT.LDREC) GOTO12
C
C          STORE THE SUBGRAPH WHOSE INDEX IS NSBR
C
   10 BREC(NSBR+1)=CMPLX(FLOAT(IADR),0.)
      BREC(NSBR)=CMPLX(REAL(BREC(NSBR)),FLOAT(LTHR))
      IAD=IADR1/2
      BREC(IAD)=CMPLX(FLOAT(N),FLOAT(NINT))
      BREC(IAD+1)=CMPLX(FLOAT(LCOL(1)),FLOAT(LES))
      N1=N+1
      DO 5 I=2,N1
    5 BREC(IAD+I)=CMPLX(FLOAT(LCOL(I)),FLOAT(LCOL(I+N)))
      N2=IAD+1+N
      N3=2*N+1
      DO 6 I=1,N
    6 BREC(N2+I)=CMPLX(FLOAT(NON(I)),FLOAT(LCOL(N3+I)))
      N2=IAD+1+2*N
      DO 7 I=1,LES
    7 BREC(N2+I)=CMPLX(FLOAT(ITAG(I)),FLOAT(ITAG(I+LES)))
   26 N3=N2+LES
      LES4=LES*4
      DO 25 I=1,LES4
   25 BREC(N3+I)=CE(I)
      NN=2*N
      N2=N3+LES4
      N3=N2+NN
      DO 8 I=1,NN
    8 BREC(N2+I)=DE(I)
   30 DO 31  I=1,N
   31 BREC(N3+I)=V(I)
C
C          STORE ADDRESS FOR THE SUBGRAPH NSUB(NBS) IN THE RECORD
C          WHOSE INDEX IS NADR
C
      NADR=(NSUB(NBS)-1)/LADR+1
      IADR=NSUB(NBS)-(NADR-1)*LADR
      IF(INDX(5).EQ.NADR) GOTO 11
      IF(IN(23).EQ.2) RETURN
      IF(NADR.GT.((IN(9)-1)/IN(12)+1)) GOTO 16
C
C          ADDRESSES FOR SUBGRAPHS ARE STORED IN RECORD NO. NADR IN
C
C          RANDOM FILE NO.5. THE INDEX OF THE BASIC RECORD WHICH STORES
C
C          THE RESULT OF THE ANALYSIS OF THE SUBGRAPH IS STORED IN THE
C
C          POSITION IADR, WHILE THE INDEX OF THE SUBGRAPH IN THE BASIC
C
C          RECORD IS STORED IN THE POSITION IADR+LADR
C
```

```
      IF(IN(23).NE.1) CALL WRITMS(5,IADREC,LLDR,INDX(5))
      IF(IN(23).EQ.1) CALL WRITMS(5,IADREC,LLDR,INDX(5),1,1)
      IF(INDX(5).GT.IN(24)) IN(24)=INDX(5)
      IF(NADR.GT.IN(24)) GOTO 18
      CALL READMS(5,IADREC,LLDR,NADR)
   18 INDX(5)=NADR
   11 IADREC(IADR)=INDX(3)
      IADREC(IADR+LADR)=NSBR
      RETURN
C
C          THIS PART IS EXECUTED AFTER UPDATING THE COEFFICIENT MATRIX
C
C          WITHIN THE SAME PATTERN OF NONZERO ELEMENTS.
C
C          READ SUBGRAPH NSUB(NSB) FROM THE BASIC RECORD
C
   21 CALL RSBR(NSUB(NBS),LADR,INDX,IADREC,BREC,LREC,LCOL,NON,DE,V,ITAG,
     +CE,INDEX3,INDEX5,N,NINT,LES,IAA,IN,IAD,NSBR)
      LES2=LES*2
      DO 27 I=1,LES2
   27 CE(I)=CE(LES2+I)
      DO 34 I=1,N
   34 DE(I)=DE(N+I)
C
C          UPDATE THE COEFFICIENT MATRIX
C
      DO 22 I=1,NEL
      NRW=NROW(I)
      NCL=NCOL(I)
      COEFF=AK(I)
      IF(NRW.GT.N.OR.NRW.LE.0.OR.NCL.GT.N.OR.NCL.LE.0) GOTO 23
      CALL CHNGE(IEX,NRW,NCL,COEFF,LCOL,ITAG,ITAG(LES+1),CE,CE(LES+1),DE
     +)
   22 IF(IEX.EQ.1) GOTO 23
C
C          STORE UPPER AND LOWER PARTS
C
      DO 32 I=1,LES2
   32 CE(LES2+I)=CE(I)
C
C          STORE DIAGONAL ELEMENTS
C
      DO 33 I=1,N
   33 DE(N+I)=DE(I)
C
C          PERFORM PARTIAL BIFACTORIZATION
C
      CALL REDUCN(N,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),CE,CE(LES+1),DE,NI
     +NT)
C
C          MULTIPLY LEFT PART OF LR BIFACTORIZATION BY THE RHS VECTOR
C
      CALL LMIRHS(NINT,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),CE,DE,V)
      N2=IAD+1+2*N
      GOTO 26
   12 WRITE(6,100)
  100 FORMAT(1H ,/," BASIC RECORD TOO SMALL TO STORE RESULTS",/,
     +" INCREASE IN(11)",/)
      IN(3)=1
      RETURN
C
C          THIS PART IS EXECUTED AFER UPDATING RHS ONLY
C
   28 N=NSUB(NMS+NBS)
      WRITE(6,29) NSUB(NBS)
```

```
   29 FORMAT(1H ,/," UPDATED RIGHT HAND SIDE VECTOR OF THE SUBGRAPH NO
     +",I3/)
      CALL PRINT(N,V)
C
C         READ SUBGRAPH NSUB(NBS) FROM THE BASIC RECORD
C
   19 CALL RSBR(NSUB(NBS),LADR,INDX,IADREC,BREC,LREC,LCOL,NON,DE,V,ITAG,
     +CE,INDEX3,INDEX5,N,NINT,LES,IAA,IN,IAD,NSBR)
C
C         MULTIPLY LEFT PART OF LR BIFACTORIZATION BY THE RHS VECTOR
C
      CALL LMIRHS(NINT,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),CE,DE,V)
      N3=IAD+1+N*4+LES*5
      GOTO 30
   14 WRITE(6,110)
  110 FORMAT(1H ,/," NUMBER OF PREDICTED BASIC RECORDS TOO SMALL",/,
     +" INCREASE IN(15)",/)
      IN(3)=1
      RETURN
   16 WRITE(6,120)
  120 FORMAT(1H ,/," NUMBER OF ADDRESSING RECORDS TOO SMALL",/,
     +" CHECK INDICES OF MODIFIED SUBGRAPHS AT THE RANDOM FILE NO 2",/)
      IN(3)=1
      RETURN
   23 WRITE(6,24)
   24 FORMAT(1H ," ELEMENT OUT OF THE MATRIX AREA",/)
      IN(3)=1
      RETURN
      END
C
C
      SUBROUTINE LMIRHS(NINT,LCOL,NSEQ,ITAG,LNXT,CE,DE,V)
C
C         THIS SUBROUTINE MULTIPLIES THE LEFT PART OF LR FACTORIZATION
C
C         BY THE RHS VECTOR FOR INTERNAL NODES OF A SUBGRAPH.
C
C         THE LEFT-HAND FACTOR MATRIX IS ASSUMED TO HAVE DIAGONAL
C
C         ELEMENTS EQUAL TO 1.
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),DE(1),V(1),CF
C
C         NINT - NUMBER OF INTERNAL NODES
C
      IF(NINT.LE.0) RETURN
      DO 20 J=1,NINT
      K=NSEQ(J)
      CF=DE(K)*V(K)
      V(K)=CF
      L=LCOL(K)
   10 IF(L.LE.0) GOTO 20
      I=ITAG(L)
      V(I)=V(I)-CE(L)*CF
      L=LNXT(L)
      GOTO 10
   20 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE ANSUBSN(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,NSUB,INDEX3,IN
     +DEX5,AK,NCOL,NROW,INDX,NON,NEXTN,RHSR,INDEX1,IT)
C
```

```
C
C       THIS SUBROUTINE PERFORMS ANALYSIS OF SUBSTITUTE GRAPHS.
C
C       ONLY SUBSTITUTE GRAPHS WHICH ARE NECESSARY FOR UPDATING THE
C
C       RESULTS OF THE PREVIOUS SOLUTION WILL BE ANALYSED
C
      DIMENSION IN(1),LCOL(1),ITAG(1),IADREC(1),NSUB(1),INDEX1(1),
     +INDEX3(1),INDEX5(1),NCOL(1),NROW(1),INDX(1),NON(1),NEXTN(1),IT(1)
      COMPLEX CE(1),DE(1),V(1),BREC(1),AK(1),RHSR(1)
C
C       NEXTN   CONTAINS NUMBERS OF EXTERNAL NODES FOR EACH SUBSTITUTE
C
C               SUBGRAPH. SUBSTITUTE SUBGRAPH NO.1 MUST HAVE ZERO
C
C               EXTERNAL NODES
C
C       CALCULATE THE ADDRESSES FOR WORKING AREAS
C
      IA1=IN(16)+1
      IA2=IN(17)/2+1
      IA3=IN(17)+1
      IA4=IN(17)+IA2
      IA5=IN(26)/IN(25)+1
      IB1=IN(16)/2+1
      LADR=IN(12)
      NSBR=IN(14)
      LREC=IN(11)
      LDRR=2*IN(25)
      LDREC=2*LREC
      LLDR=2*LADR
      IF(IN(23).EQ.0.OR.IN(23).EQ.2) GOTO 8
C
C       READ STORED PARTIAL SOLUTION
C
      IF(IN(23).EQ.3)GOTO 12
      DO 11 I=1,IA5
      CALL READMS(1,RHSR,LDRR,IA5+I)
   11 CALL WRITMS(1,RHSR,LDRR,I,1,1)
   12 CONTINUE
      IF(IA5.NE.INDX(4)) CALL READMS(1,RHSR,LDRR,INDX(4))
    8 CONTINUE
C
C       READ NUMBERS OF EXTERNAL NODES FOR THE SUBSTITUTE SUBGRAPHS
C
      ME=IN(9)/2
      DO 9 I=1,ME
    9 NEXTN(I)=IN(26+I)
C
C       FIND A SEQUENCE OF MODIFIED SUBSTITUTE SUBGRAPHS
C
      NMS=IN(10)
      IORD=0
      CALL SMSSUB(NMS,NSUB,NCHS,IORD)
C
C       ANALYSE THE MODIFIED SUBSTITUTE SUBGRAPHS
C
      NS=NMS+1
      IF(IN(7).EQ.1.OR.IN(7).EQ.3) GOTO 2
      WRITE(6,100)(NSUB(I),I=NS,NCHS)
  100 FORMAT(1H ,/," SEQUENCE OF SUBSTITUTE SUBGRAPHS TO BE REANALYZED"
     +,//(20I4))
    2 CONTINUE
C
C       MODIFIED SUBSTITUTE SUBGRAPHS HAVE INDICES PLACED IN NSUB
C
```

```
C          IN ADDRESSES FROM NS+1 TO NCHS. THE HIGHER ADDRESS
C
C          THE LOWER THE INDEX OF THE CORRESPONDING SUBSTITUTE SUBGRAPH.
C
      DO 1 I=NS,NCHS
      NSB=NSUB(I)
      NSB1=2*NSB
      NSB2=NSB1+1
      CALL CHEBSUB(NSB1,IT,IR1)
      CALL CHEBSUB(NSB2,IT,IR2)
      IF(IR1.EQ.0.OR.IR2.EQ.0) GOTO 1
      CALL ABSUB(NSB,IT)
      CALL CONNECT(NSB1,NSB2,N,NEL,LCOL(1),LCOL(IA1),ITAG(1),ITAG(IA2)
     +,ITAG(IA3),ITAG(IA4),CE(1),CE(IA3),DE(1),DE(IB1),NON,NROW,NCOL,AK,
     +V,BREC,INDX,IADREC,LADR,IN,LREC,INDEX3,INDEX5)
      IF(IN(3).EQ.1) RETURN
      IN(1)=N
      IN(2)=NEL
      IN(3)=2
      IN(6)=N-NEXTN(NSB)
C
C          CHECK DIAGONAL ELEMENTS IN THE SUBMATRIX
C
      CALL CHEDEL(IN(6),NEL,NCOL,NROW,NSUB,NSB,LCOL,IN(3))
      IF(IN(3).EQ.1)RETURN
      IF(IN(6).GT.0) GOTO 3
      WRITE(6,110)
  110 FORMAT(1H ,/," SUBSTITUTE MATRIX OF ZERO DIMENSION",/,"CHECK DATA
     +DESCRIBING INDICES OF EXTERNAL NODES FOR SUBSTITUTE SUBGRAPHS")
      IN(3)=1
      RETURN
    3 IN3=0
      CALL ASUBSR(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,NSUB,INDEX3,
     +INDEX5,AK,NCOL,NROW,INDX,NON,N,NEL,I,NSBR,LREC,LADR,IN3)
C
C          CALCULATE THE ADDRESS AND STORE RHS VECTOR
C
      CALL CASRHS(IN,RHSR,NSB,INDX,INDEX1,N,NEXTN,NSUB,V,NON)
      IF(IN(3).EQ.1)RETURN
    1 CONTINUE
      IF(IN(23).EQ.2) RETURN
C
C          STORE FINAL RESULTS (NOT NECESSARY WHEN THE PROGRAM IS
C          EXECUTED WITHOUT CREATING PERMANENT FILES)
C
      LLD1=LADR*2
      IN(13)=IN(13)+1
      IF(IN(23).EQ.1) IN(13)=INDX(3)
      INDX(3)=IN(13)
      IF(IN(13).GT.IN(15)) GOTO 5
      IF(INDX(5).GT.((IN(9)-1)/LADR+1)) GOTO 7
      IF(IN(23).EQ.0.OR.IN(23).EQ.3) GOTO 4
      CALL WRITMS(1,RHSR,LDRR,IN(21),1,1)
      CALL WRITMS(3,BREC,LDREC,IN(13),1,1)
      CALL WRITMS(5,IADREC,LLDR,INDX(5),1,1)
      DO 6 I=1,IA5
      CALL READMS(1,RHSR,LDRR,I)
    6 CALL WRITMS(1,RHSR,LDRR,IA5+I,1,1)
      RETURN
    4 CALL WRITMS(1,RHSR,LDRR,IN(21))
      CALL WRITMS(3,BREC,LDREC,IN(13))
      CALL WRITMS(5,IADREC,LLDR,INDX(5))
      DO 10 I=1,IA5
      CALL READMS(1,RHSR,LDRR,I)
   10 CALL WRITMS(1,RHSR,LDRR,IA5+I)
```

```
      RETURN
    5 WRITE(6,120)
  120 FORMAT(1H ,/," NUMBER OF PREDICTED BASIC RECORDS TOO SMALL",/,
     +" INCREASE IN(15)",/)
      IN(3)=1
      RETURN
    7 WRITE(6,140)
  140 FORMAT(1H ,/," NUMBER OF ADDRESSING RECORDS TOO SMALL ",/," CHECK
     +INDICES OF MODIFIED SUBGRAPHS AT THE RANDOM FILE NO 2",/)
      IN(3)=1
      RETURN
      END
C
C
      SUBROUTINE SMSSUB(NMS,NSUB,NCHS,IORD)
C
C         THIS SUBROUTINE DETERMINES A SEQUENCE OF MODIFIED SUBSTITUTE
C
C         SUBGRAPHS TO BE REANALYZED. NCHS IS THE TOTAL NUMBER OF
C
C         SUBGRAPHS WHICH ARE MODIFIED DURING THE HIERARCHICAL
C
C         ANALYSIS. INDICES OF SUBSTITUTE SUBGRAPHS MODIFIED
C
C         DURING THE ANALYSIS ARE STORED IN THE MATRIX NSUB IN
C
C         ADDRESSES FROM NMS+1 TO NCHS
C
       INTEGER NSUB(1)
C
C         ORDER MODIFIED SUBGRAPHS IN DECREASING SEQUENCE
C
       IF(NMS.LE.1) GOTO 2
       DO 1 I=2,NMS
       J=I
    3  IF(NSUB(J).LT.NSUB(J-1)) GOTO 1
       IA=NSUB(J)
       NSUB(J)=NSUB(J-1)
       NSUB(J-1)=IA
       J=J-1
       IF(J.LE.1) GOTO 1
       GOTO 3
    1  CONTINUE
    2  CONTINUE
C
C         PREVIOUS PART OF NSUB CONTAINING LENGTHS OF RECORDS
C         STORING INFORMATION ABOUT BLOCKS WILL BE CHANGED
C
       I=1
       J=NMS
    6  IA=NSUB(I)
       IB=IA/2
       IF(IB.EQ.0) GOTO 4
       IF(I.GT.NMS) GOTO 9
       I=I+1
       IF(IA.EQ.IB*2) GOTO 11
    5  IC=NSUB(I)/2
       IF(IB.NE.IC) GOTO 11
       I=I+1
       GOTO 11
    7  J=J+1
       NSUB(J)=IB
       GOTO 6
    9  I=I+1
   11  IF(I.GT.J)GOTO 7
```

```
      DO 10 IX=I,J
      IF(IB.NE.NSUB(IX)) GOTO 10
      GOTO 6
   10 CONTINUE
      GOTO 7
    4 NCHS=J
C
C          ORDER MODIFIED SUBGRAPHS IN DECREASING SEQUENCE
C
      IF(NCHS.LE.1) RETURN
      DO 14 I=2,NCHS
      J=I
   13 IF(NSUB(J).LT.NSUB(J-1)) GOTO 14
      IA=NSUB(J)
      NSUB(J)=NSUB(J-1)
      NSUB(J-1)=IA
      J=J-1
      IF(J.LE.1) GOTO 14
      GOTO 13
   14 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CONNECT(NSB1,NSB2,N,NEL,NON1,NON2,NROW1,NROW2,NCOL1,NCO
     +L2,AK1,AK2,V1,V2,NON,NROW,NCOL,AK,V,BREC,INDX,IADREC,LADR,IN,LREC
     +,INDEX3,INDEX5)
C
C          THIS SUBROUTINE COMBINES THE RESULTS OF ANALYSIS OF TWO
C
C          SUBGRAPHS NSUB1 AND NSUB2 INTO THE DESCRIPTION OF A PROPER
C
C          SUBGRAPH SPANNED OVER THEIR EXTERNAL NODES
C
      DIMENSION NON1(1),NON2(1),NROW1(1),NROW2(1),NCOL1(1),NCOL2(1),NON(
     +1),NROW(1),NCOL(1),INDX(1),IADREC(1),IN(1),INDEX3(1),INDEX5(1)
      COMPLEX AK1(1),AK2(1),V1(1),V2(1),AK(1),V(1),BREC(1)
C
C          N          IS THE NUMBER OF NODES IN THE SUBSTITUTE SUBGRAPH
C
C          NEL        IS THE NUMBER OF NONZERO ELEMENTS IN THE SUBMATRIX
C                     REPRESENTING THE SUBSTITUTE SUBGRAPH
C
C          NROW,NCOL DESCRIBE ROW AND COLUMN INDICES OF ELEMENTS IN THE
C                     RESULTING SUBGRAPH
C
C          AK         COEFFICIENT MATRIX OF DIMENSION NOT LESS THAN
C                     NEL1+NEL2
C
C          V          RHS VECTOR OF DIMENSION NOT LESS THAN N
C
      CALL EEDSUB(NSB1,N1,NEL1,NROW1,NCOL1,NON1,BREC,AK1,V1,INDX,IADREC,
     +LADR,LREC,INDEX3,INDEX5,IN)
      IF(IN(3).EQ.1) RETURN
      IF(IN(7).NE.2)GOTO 1
      WRITE(6,100)NSB1,N1,NEL1
      CALL PRINTNC(NEL1,NROW1,NCOL1,AK1)
  100 FORMAT(1H ,/," EXTERNAL DESCRIPTION OF THE SUBGRAPH NO",I3,/
     +," NUMBER OF NODES:",I3,4X," NUMBER OF ELEMENTS:",I3,/,)
    1 CONTINUE
      CALL EEDSUB(NSB2,N2,NEL2,NROW2,NCOL2,NON2,BREC,AK2,V2,INDX,IADREC,
     +LADR,LREC,INDEX3,INDEX5,IN)
      IF(IN(3).EQ.1) RETURN
      IF(IN(7).NE.2) GOTO 2
      WRITE(6,100)NSB2,N2,NEL2
```

```
      CALL PRINTNC(NEL2,NROW2,NCOL2,AK2)
    2 CONTINUE
      CALL CONNUMB(NON1,NON2,NON,N1,N2,N)
      CALL ADDSUB(N1,N2,N,NEL1,NEL2,NEL,NROW1,NROW2,NROW,NCOL1,NCOL2,
     +NCOL,NON1,NON2,AK1,AK2,AK,V1,V2,V)
      RETURN
      END
C
C
      SUBROUTINE EEDSUB(NSUB,NEX,NEL,NROW,NCOL,NON,BREC,AK,V,INDX,
     +IADREC,LADR,LREC,INDEX3,INDEX5,IN)
C
C         THIS SUBROUTINE EXTRACTS EXTERNAL DESCRIPTION OF THE SUB-
C
C         GRAPH WHOSE INDEX IS NSUB FROM THE DESCRIPTION OF THE SUBGRAPH
C
C         AT THE LOWER LEVEL.
C
C         AT THE OUTPUT THE SUBGRAPH IS DESCRIBED BY THE COEFFICIENT
C
C         MATRIX AK, ITS ROWS AND COLUMNS BY MATRICES NROW & NCOL,
C
C         RESPECTIVELY. ROW AND COLUMN INDICES ARE WITHIN THE RANGE
C
C         1 TO NEX. NEX IS THE NUMBER OF NODES.
C
C         NEL IS THE NUMBER OF NONZERO ELEMENTS
C
      INTEGER NON(1),NROW(1),NCOL(1),INDX(1),IADREC(1),INDEX3(1),INDEX5(
     +1),IN(1)
      COMPLEX AK(1),BREC(1),V(1),REE
C
C         FIND THE ADDRESS FOR THE SUBGRAPH NSUB
C
      LDREC=2*LREC
      LLDR=LADR*2
      NADR=(NSUB-1)/LADR+1
      IADR=NSUB-(NADR-1)*LADR
      IF(INDX(5).EQ.NADR) GOTO 4
      IF(NADR.GT.((IN(9)-1)/IN(12)+1)) GOTO 6
C
C         STORE OLD ADDRESSING RECORD
C
      IF(IN(23).NE.1) CALL WRITMS(5,IADREC,LLDR,INDX(5))
      IF(IN(23).EQ.1) CALL WRITMS(5,IADREC,LLDR,INDX(5),1,1)
      IF(INDX(5).GT.IN(24)) IN(24)=INDX(5)
      CALL READMS(5,IADREC,LLDR,NADR)
      INDX(5)=NADR
    4 NRBR=IADREC(IADR)
      NSBR=IADREC(IADR+LADR)
C
C         READ THE DESCRIPTION OF THE SUBGRAPH
C
      IF(INDX(3).EQ.NRBR) GOTO 5
      IF(INDX(3).GT.IN(15)) GOTO 7
C
C         STORE OLD BASIC RECORD
C
      IF(IN(23).NE.1) CALL WRITMS(3,BREC,LDREC,INDX(3))
      IF(IN(23).EQ.1) CALL WRITMS(3,BREC,LDREC,INDX(3),1,1)
      CALL READMS(3,BREC,LDREC,NRBR)
      INDX(3)=NRBR
    5 IADR=REAL(BREC(NSBR))
      LTHR=AIMAG(BREC(NSBR))
      IAD=IADR/2
```

```
      N=REAL(BREC(IAD))
      NINT=AIMAG(BREC(IAD))
      LES=AIMAG(BREC(IAD+1))
C
C         CALCULATE MATRICES AK,V,NROW,NCOL,NON
C
      IA=IAD+NINT
      IA1=IA+N+1
      IA4=IA1+5*LES+N
      IA5=IA4+2*N
      NEX=N-NINT
      DO 1 I=1,NEX
      AK(I)=BREC(IA4+I)
      V(I)=BREC(IA5+I)
      NROW(I)=I
      NCOL(I)=I
    1 NON(I)=REAL(BREC(IA1+I))
C
C         CALCULATE MATRICES AK,NROW,NCOL
C
      IA1=IAD+2*N+1
      IA2=IA1+LES
      IA3=IA2+LES
      J=NEX
      DO 2 I=1,NEX
      IS=REAL(BREC(IA+I))
    3 IF(IS.LE.0) GOTO 2
      J=J+1
      AK(J)=BREC(IA2+IS)
      IRR=REAL(BREC(IA1+IS))-NINT
      NROW(J)=IRR
      NCOL(J)=I
      J=J+1
      AK(J)=BREC(IA3+IS)
      NROW(J)=I
      NCOL(J)=IRR
C
C         ADDRESS FOR THE NEXT ELEMENT
C
      IS=AIMAG(BREC(IA1+IS))
      GOTO 3
    2 CONTINUE
      NEL=J
      RETURN
    6 WRITE(6,120)
  120 FORMAT(1H ,/," NUMBER OF ADDRESSING RECORDS TOO SMALL",/," CHECK N
     +UMBERS OF MODIFIED SUBGRAPHS IN THE RANDOM FILE NO 2",/)
      IN(3)=1
      RETURN
    7 WRITE(6,130)
  130 FORMAT(1H ,/," NUMBER OF PREDICTED BASIC RECORDS TOO SMALL",/,
     +" INCREASE IN(15)",/)
      IN(3)=1
      RETURN
      END
C
C
      SUBROUTINE CONNUMB(NON1,NON2,NON,N1,N2,N)
C
C         THIS SUBROUTINE ASSIGNS ORIGINAL INDICES OF THE NODES TO THE
C
C         NODES OF CONNECTED SUBGRAPHS. INPUT INFORMATION ABOUT THE
C
C         ORIGINAL INDICES OF SUBGRAPH NODES IS PLACED IN NON1 AND NON2.
C
```

```
C
C        ON OUTPUT NON1 AND NON2 CONTAIN CURRENT INDICES OF
C
C        SUBGRAPHS NODES IN THE RESULTING SUBSTITUTE GRAPH.
C
      INTEGER NON1(1),NON2(1),NON(1)
C
C        ASSIGN INDICES TO THE NODES OF THE SUBSTITUTE GRAPH.
C
C        N IS THE NUMBER OF NODES IN THE SUBSTITUTE SUBGRAPH
C
C        NON ARE ORIGINAL INDICES OF THE NODES IN THE SUBSTITUTE
C
C        SUBGRAPH
C
      I1=1
      I2=1
      I=0
      LN1=NON1(I1)
      LN2=NON2(I2)
    4 IF(LN1-LN2)1,2,3
    1 I=I+1
      NON(I)=LN1
      I1=I1+1
      IF(I1.GT.N1) GOTO 5
      LN1=NON1(I1)
      GOTO 4
    2 I=I+1
      NON(I)=LN1
      I1=I1+1
      I2=I2+1
      LN1=NON1(I1)
      LN2=NON2(I2)
      IF(I1.GT.N1) GOTO 5
      IF(I2.GT.N2) GOTO 6
      GOTO 4
    3 I=I+1
      NON(I)=LN2
      I2=I2+1
      IF(I2.GT.N2) GOTO 6
      LN2=NON2(I2)
      GOTO 4
    5 IF(I2.GT.N2) GOTO 7
      I=I+1
      NON(I)=LN2
      I2=I2+1
      LN2=NON2(I2)
      GOTO 5
    6 IF(I1.GT.N1) GOTO 7
      I=I+1
      NON(I)=LN1
      I1=I1+1
      LN1=NON1(I1)
      GOTO 6
    7 CONTINUE
      N=I
C
C        RENUMBER THE EXTERNAL NODES OF CONNECTED SUBGRAPHS
C
      J=1
      DO 8 I=1,N1
      NL=NON1(I)
   10 IF(NON(J).LT.NL) GOTO 9
      NON1(I)=J
      J=J+1
    8 CONTINUE
```

```
      GOTO 14
    9 J=J+1
      GOTO 10
   14 J=1
      DO 11 I=1,N2
      NL=NON2(I)
   13 IF(NON(J).LT.NL) GOTO 12
      NON2(I)=J
      J=J+1
   11 CONTINUE
      RETURN
   12 J=J+1
      GOTO 13
      END
C
C
      SUBROUTINE ADDSUB(N1,N2,N,NEL1,NEL2,NEL,NROW1,NROW2,NROW,NCOL1,NCO
     +L2,NCOL,NON1,NON2,AK1,AK2,AK,V1,V2,V)
C
C         THIS SUBROUTINE ADDS TWO SUBMATRICES DESCRIBING SUBSTITUTE
C
C         SUBGRAPHS TO OBTAIN A MATRIX DESCRIBING THEIR INTERCONNECTION
C
      INTEGER NROW1(1),NROW2(1),NROW(1),NCOL1(1),NCOL2(1),NCOL(1),NON1(1
     +),NON2(1)
      COMPLEX AK1(1),AK2(1),AK(1),V1(1),V2(1),V(1)
C
C         N              IS THE NUMBER OF NODES IN THE SUBSTITUTE SUBGRAPH
C
C         NEL1,NEL2,NEL ARE THE INDICES OF NONZERO ELEMENTS IN
C                        SUBMATRICES TO BE ADDED AND IN THE RESULTING
C                        SUBMATRIX, RESPECTIVELY
C
C         NROW,NCOL DESCRIBE ROW AND COLUMN INDICES OF ELEMENTS IN
C                        RESULTING SUBMATRIX
C
C         AK             THE COEFFICIENT MATRIX
C
C         V              RHS VECTOR
C
      NX=NEL1+NEL2
      DO 1 I=1,NX
    1 AK(I)=CMPLX(0.,0.)
      DO 2 I=1,N
    2 V(I)=CMPLX(0.,0.)
C
C         CALCULATE DIAGONAL ELEMENTS AND RHS
C
      I1=1
      I2=1
      DO 3 I=1,N
      L1=NON1(I1)
      L2=NON2(I2)
      IF(L1.EQ.I) GOTO 4
    5 IF(I2.GT.N2)GOTO 3
      AK(I)=AK2(I2)+AK(I)
      V(I)=V(I)+V2(I2)
      NROW(I)=I
      NCOL(I)=I
      I2=I2+1
      GOTO 3
    4 IF(I1.GT.N1) GOTO 5
      AK(I)=AK(I)+AK1(I1)
      V(I)=V(I)+V1(I1)
      NROW(I)=I
```

```
      NCOL(I)=I
      I1=I1+1
      IF(L2.EQ.I) GOTO 5
    3 CONTINUE
C
C         CALCULATE THE REMAINING PART OF THE COEFFICIENT MATRIX
C
      I=N+1
      NS1=I1
      NS2=I2
      DO 6  JJ=NS1,NEL1
      IR1=NON1(NROW1(JJ))
      IC1=NON1(NCOL1(JJ))
      AK(I)=AK(I)+AK1(JJ)
      NROW(I)=IR1
      NCOL(I)=IC1
      DO 7 J3=NS2,NEL2
      IR2=NON2(NROW2(J3))
      IC2=NON2(NCOL2(J3))
      IF(IR1.EQ.IR2.AND.IC1.EQ.IC2)AK(I)=AK(I)+AK2(J3)
    7 CONTINUE
      I=I+1
    6 CONTINUE
      DO 8 J3=NS2,NEL2
      IR2=NON2(NROW2(J3))
      IC2=NON2(NCOL2(J3))
      DO 9 JJ=NS1,NEL1
      IR1=NON1(NROW1(JJ))
      IC1=NON1(NCOL1(JJ))
      IF(IR1.EQ.IR2.AND.IC1.EQ.IC2)GOTO 8
    9 CONTINUE
      AK(I)=AK(I)+AK2(J3)
      NROW(I)=IR2
      NCOL(I)=IC2
      I=I+1
    8 CONTINUE
   14 NEL=I-1
      RETURN
      END
C
C
      SUBROUTINE CASRHS(IN,RHSR,NSB,INDX,INDEX1,N,NEXTN,NSUB,V,NON)
C
C         THIS SUBROUTINE CALCULATES ADDRESSES AND STORES THE RHS VECTOR
C
C         FOR THE SUBSTITUTE SUBGRAPH NSB
C
      INTEGER IN(1),INDX(1),INDEX1(1),NEXTN(1),NSUB(1),NON(1)
      COMPLEX RHSR(1),V(1)
C
C          RHSR        RIGHT HAND SIDE RECORD
C
C          NRHSR       INDEX OF RIGHT HAND SIDE RECORD
C
C          IAD1        ADDRESS IN THE RIGHT HAND SIDE RECORD, WHERE THE
C                      INFORMATION ABOUT GIVEN SUBSTITUTE SUBGRAPH BEGINS
C
C          NEXTN       NUMBERS OF EXTERNAL NODES FOR THE SUBSTITUTE
C                      SUBGRAPHS. SUBSTITUTE SUBGRAPH NO.1 HAS ZERO
C                      EXTERNAL NODES
C
C          INDX(4)     CURRENT INDEX OF THE RHS RECORD
C
C          LRHSR       LENGTH OF RHS RECORD
C
```

```
      IN25=IN(25)
      LADR=IN(12)
      LLD1=LADR*2
      LRHS=2*IN25
C
C         FIND THE SMALLEST EXTERNAL NODE IN NSB
C
      NSEN=NON(1)-IN(19)+1
      NRHSR=NSEN/IN25+1
C
C         STORE OLD RHS RECORD
C
      IF(NRHSR.EQ.INDX(4)) GOTO 9
      IF(IN(23).NE.1) CALL WRITMS(1,RHSR,LRHS,IN(21))
      IF(IN(23).EQ.1) CALL WRITMS(1,RHSR,LRHS,IN(21),1,1)
      CALL READMS(1,RHSR,LRHS,NRHSR)
      INDX(4)=NRHSR
      IN(21)=NRHSR
    9 IAD1=NSEN-(NRHSR-1)*IN25
C
C         CALCULATE THE ADDRESS IN THE RHS RECORD
C
      IAD=IAD1+N-NEXTN(NSB)-1
      IF(IAD.LT.IAD1) RETURN
      IJ=0
      IF(IAD.LE.IN25) GOTO 8
    6 DO 10 I=IAD1,IN25
      IJ=IJ+1
   10 RHSR(I)=V(IJ)
C
C         STORE THE RHS RECORD
C
      IF(IN(23).NE.1) CALL WRITMS(1,RHSR,LRHS,IN(21))
      IF(IN(23).EQ.1) CALL WRITMS(1,RHSR,LRHS,IN(21),1,1)
      IN(21)=IN(21)+1
      INDX(4)=IN(21)
      IAD1=1
      IAD=IAD-IN25
      IF(IAD.GT.IN25) GOTO 6
C
C         WRITE THE RHS VECTOR ASSOCIATED WITH THE SUBGRAPH.
C
C         THE ELEMENT RHSR(IAA), WHERE IAA=NSEN-(NRHSR-1)*IN(25),
C
C         STORES THE RHS ELEMENT CORRESPONDING TO NON(1), WHILE THE RHS
C
C         ELEMENT FOR THE NODE NON(I) WILL BE AT THE ADDRESS
C
C         NON(I)-NON(1)+IAA
C
    8 DO 11 I=IAD1,IAD
      IJ=IJ+1
   11 RHSR(I)=V(IJ)
      RETURN
      END
C
C
      SUBROUTINE PSHDS(IN,LCOL,ITAG,CE,DE,V,IADREC,BREC,NSUB,INDEX3,INDE
     +X5,INDX,NON,INDEX1,INDEX7,RHSR,SOLR,IMD)
C
C         THIS SUBROUTINE CALCULATES A PARTIAL SOLUTION FOR THE
C
C         HIERARCHICALLY DECOMPOSED SYSTEM OF LINEAR EQUATIONS
C
      INTEGER NSUB(1),NON(1),INDX(1),IADREC(1),LCOL(1),ITAG(1),INDEX3(1)
```

```
     +,INDEX1(1),INDEX5(1),INDEX7(1),IN(1),IM(1)
      COMPLEX BREC(1),V(1),DE(1),CE(1),RHSR(1),SOLR(1)
C
      LREC=IN(11)
      NMS=IN(10)
      IRHSUF=IN(22)
      LL=0
C
C         IRHSUF     INDICATOR FOR RHS UPDATING FORMULA
C                  =0 COMPLETE RHS VECTOR WILL BE RECALCULATED
C                  =1 ONLY THE RHS FOR THE MODIFIED SUBGRAPHS WILL BE
C                     RECALCULATED
C                  =2 ONLY THE RHS FOR THE SUBGRAPHS SPECIFIED BY THE USER
C                     WILL BE RECALCULATED
C                  =3 COMPLETE RHS WILL BE RECALCULATED
C
      IF(IRHSUF.EQ.1) GOTO 9
      IF(IRHSUF.EQ.2) GOTO 2
C
C         COMPLETE RHS WILL BE RECALCULATED
C
      NMS=IN(5)
      NREC=26+IN(9)/2
      GOTO 3
C
C         RHS WILL BE RECALCULATED FOR THE SUBGRAPHS SPECIFIED
C         BY THE USER
C
    2 NMS=IN(4)
      NREC=26+IN(9)/2+IN(5)
    3 DO 4 I=1,NMS
    4 NSUB(I)=IN(NREC+I)
      IORD=1
    9 CALL SMSSUB(NMS,NSUB,NCHS,IORD)
      IF(IN(7).EQ.1.OR.IN(7).EQ.3) GOTO 1
      WRITE(6,100)(NSUB(I),I=1,NCHS)
  100 FORMAT(1H ,/," SOLUTION WILL BE RECALCULATED AT THE INTERNAL NODES
     + OF SUBGRAPHS",//(20I4))
    1 CONTINUE
C
C         MULTIPLY RIGHT PART OF LR FACTORIZATION BY THE RHS VECTOR
C
      IF(IN(7).EQ.1) GOTO 7
      WRITE(6,110)
  110 FORMAT(1H ,/," SOLUTION",/,1H ,"________",/)
    7 CONTINUE
      DO 5 IJ=1,NCHS
      NRS=NCHS-IJ+1
      CALL RMRHS(NRS,NMS,NSUB,IN,INDX,IADREC,BREC,LREC,LCOL,NON,DE,V,ITA
     +G,CE,INDEX3,INDEX5,RHSR,IM,LL,SOLR,INDEX7,INDEX1)
      IF(IN(3).EQ.1) RETURN
    5 CONTINUE
C
C         STORE FINAL RESULTS
C
      IM5=IM(5)
      IM(5)=LL
      IF(IN(7).NE.1)WRITE(6,120)LL
  120 FORMAT(1H ,/," NUMBER OF VARIABLES IN THE SOLUTION VECTOR",I4)
      IF(IN(23).EQ.2)RETURN
      LSOL=4*IM5
      IF(IN(23).EQ.0.OR.IN(23).EQ.3) GOTO 6
      CALL WRITMS(7,SOLR,LSOL,INDX(7),1,1)
      RETURN
    6 CALL WRITMS(7,SOLR,LSOL,INDX(7))
```

```
      RETURN
      END
C
C
      SUBROUTINE RMRHS(NRS,NMS,NSUB, IN, INDX, IADREC, BREC, LREC, LCOL, NON,
     +DE, V, ITAG, CE, INDEX3, INDEX5, RHSR, IM, LL, SOLR, INDEX7, INDEX1)
C
C          THIS SUBROUTINE MULTIPLIES THE RIGHT PART OF THE LR
C
C          FACTORIZATION BY THE RHS VECTOR FOR (NRS)TH SUBGRAPH. THE RHS
C
C          IS NOT UPDATED FOR PROPER BLOCKS
C
      INTEGER NSUB(1),NON(1),INDX(1),IADREC(1),LCOL(1),ITAG(1),INDEX3(1)
     +,INDEX1(1),INDEX5(1),IN(1),IARRHS(1),IM(1),INDEX7(1)
      COMPLEX BREC(1),V(1),DE(1),CE(1),RHSR(1),SOLR(1)
C
C          NSB1     IS THE INDEX OF THE SUBGRAPH FOR WHICH THE RHS IS
C                   CALCULATED FOR INTERNAL NODES ONLY
C
C          NSB      IS THE INDEX OF THE SUBSTITUTE SUBGRAPH FOR WHICH THE
C                   RHS VECTOR IS CALCULATED
C
      NSB1=NSUB(NRS)
      IN25=IN(25)
      LADR=IN(12)
      LLD1=2*LADR
      LRHS=2*IN(25)
      INSS=0
C
C          READ SUBGRAPH NSB1
C
      CALL RSBR(NSB1,LADR, INDX, IADREC, BREC, LREC, LCOL, NON, DE, V, ITAG, CE, IN
     +DEX3, INDEX5, N, NINT, LES, IA4, IN, IAD, NSBR)
      DO 12 I=1,N
12    V(I)=BREC(IA4+I)
      N1=NON(NINT+1)
      NSEN=NON(1)-IN(19)+1
      NRHSR=NSEN/IN25+1
      IF(NRHSR.LT.1) NRHSR=1
    2 NSB=NSB1/2
      IF(NSB.EQ.0) GOTO 8
C
C          STORE OLD RHS RECORD
C
      IF(NRHSR.EQ.INDX(4)) GOTO 9
      IF(NRHSR.GT.(IN(26)/IN(25)+2)) GOTO 14
      IF(IN(23).NE.1) CALL WRITMS(1,RHSR,LRHS,IN(21))
      IF(IN(23).EQ.1) CALL WRITMS(1,RHSR,LRHS,IN(21),1,1)
      CALL READMS(1,RHSR,LRHS,NRHSR)
      INDX(4)=NRHSR
      IN(21)=NRHSR
    9 IAD1=NSEN-(NRHSR-1)*IN25
      N1=(NRHSR-1)*IN25+IN(19)
C
C          MULTIPLY THE RIGHT PART OF THE LR FACTORIZATION BY THE RHS
C          VECTOR FOR EXTERNAL NODES
C
      CALL RMERHS(NINT,NON,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),
     +CE(LES+1),V,RHSR,INSS,N1,IN25)
      IF(INSS.EQ.0) GO TO 8
    3 NSB1=NSB
      NRHSR=NRHSR+1
      GOTO 2
    8 NSB1=NSUB(NRS)
```

```
      IF(NRS.LE.NMS) GOTO 1
    5 NSEN=NON(1)-IN(19)+1
      NRHSR=NSEN/IN25+1
      IF(NRHSR.LT.1) NRHSR=1
C
C          STORE OLD RHS RECORD
C
      IF(NRHSR.EQ.INDX(4)) GOTO 6
      IF(NRHSR.GT.(IN(26)/IN(25)+2)) GOTO 14
      IF(IN(23).NE.1) CALL WRITMS(1,RHSR,LRHS,IN(21))
      IF(IN(23).EQ.1) CALL WRITMS(1,RHSR,LRHS,IN(21),1,1)
      CALL READMS(1,RHSR,LRHS,NRHSR)
      INDX(4)=NRHSR
      IN(21)=NRHSR
    6 IAD1=NSEN-(NRHSR-1)*IN25
C
C          MULTIPLY THE RIGHT PART OF THE LR FACTORIZATION BY THE RHS
C          VECTOR FOR INTERNAL NODES
C
    1 CALL RMIRHS(NINT,LCOL,LCOL(2*N+2),ITAG,ITAG(LES+1),CE(LES+1),V)
      IF(NRS.LE.NMS) GOTO 10
C
C          STORE RHS VECTOR
C
      DO 11 IJ=1,NINT
   11 RHSR(IAD1+IJ-1)=V(IJ)
   10 CONTINUE
C
C          CALCULATE THE ADDRESS AND STORE SOLUTION VECTOR
C
      CALL CASSOL(IN,IM,LL,SOLR,NINT,INDX,V,NON,INDEX7)
      IF(IN(3).EQ.1) RETURN
      IF(IN(7).EQ.1) RETURN
      WRITE(6,140)NSUB(NRS)
  140 FORMAT(1H ,/," IN SUBGRAPH NO.",I4,/)
      CALL PRINTRS(NINT,V,NON)
      RETURN
   14 WRITE(6,100)
  100 FORMAT(1H ,/," NUMBER OF RHS RECORDS TOO SMALL",/," INCREASE IN(19
     +)",/)
      IN(3)=1
      RETURN
      END
C
C
      SUBROUTINE RSBR(NSUB,LADR,INDX,IADREC,BREC,LREC,LCOL,NON,DE,V,ITAG
     +,CE,INDEX3,INDEX5,N,NINT,LES,IAA,IN,IAD,NSBR)
C
C          THIS SUBROUTINE READS SUBGRAPH WHOSE INDEX IS NSUB FROM
C
C          THE BASIC RECORD
C
      INTEGER NON(1),INDX(1),IADREC(1),LCOL(1),ITAG(1),INDEX3(1),INDEX5(
     +1),IN(1)
      COMPLEX BREC(1),V(1),DE(1),CE(1)
C
C          FIND THE ADDRESS FOR SUBGRAPH NSUB
C
      LDREC=2*LREC
      LLDR=LADR*2
      NADR=(NSUB-1)/LADR+1
      IADR=NSUB-(NADR-1)*LADR
      IF(INDX(5).EQ.NADR) GOTO 4
      CALL WRITMS(5,IADREC,LLDR,INDX(5),1,1)
      CALL READMS(5,IADREC,LLDR,NADR)
```

```
      INDX(5)=NADR
    4 CONTINUE
      INDX(5)=NADR
      NRBR=IADREC(IADR)
      NSBR=IADREC(IADR+LADR)
C
C          READ THE DESCRIPTION OF THE SUBGRAPH
C
      IF(INDX(3).EQ.NRBR) GOTO 3
C
C          STORE THE UPDATED BASIC RECORD
C
      IF(IN(23).NE.1) CALL WRITMS(3,BREC,LDREC,INDX(3))
      IF(IN(23).EQ.1) CALL WRITMS(3,BREC,LDREC,INDX(3),1,1)
      CALL READMS(3,BREC,LDREC,NRBR)
      INDX(3)=NRBR
    3 IADR=REAL(BREC(NSBR))
      LTHR=AIMAG(BREC(NSBR))
      IAD=IADR/2
      N=REAL(BREC(IAD))
      NINT=AIMAG(BREC(IAD))
      LES=AIMAG(BREC(IAD+1))
C
C          READ MATRICES LCOL,NON,DE,V
C
      LCOL(1)=REAL(BREC(IAD+1))
      IA1=IAD+1
      IA2=IA1+N
      IA3=IA2+N+5*LES
      IA4=IA3+2*N
      IAA=IA4
      NDB=2*N+1
      DO 1 I=1,N
      LCOL(I+1)=INT(REAL(BREC(IA1+I)))
      LCOL(N+1+I)=INT(AIMAG(BREC(IA1+I)))
      LCOL(NDB+I)=INT(AIMAG(BREC(IA2+I)))
      NON(I)=INT(REAL(BREC(IA2+I)))
      DE(I)=BREC(IA3+I)
      DE(N+I)=BREC(IA3+N+I)
    1 CONTINUE
      IF(IN(8).GT.6) GOTO 6
      DO 7 I=1,N
    7 V(I)=BREC(IA4+I)
    6 CONTINUE
C
C          READ MATRICES ITAG,CE
C
      IA3=IA2+N
      IA4=IA3+LES
      DO 2 I=1,LES
      ITAG(I)=INT(REAL(BREC(IA3+I)))
    2 ITAG(LES+I)=INT(AIMAG(BREC(IA3+I)))
      LES4=LES*4
      DO 5 I=1,LES4
    5 CE(I)=BREC(IA4+I)
      RETURN
      END
C
C
      SUBROUTINE RMERHS(NINT,NON,LCOL,NSEQ,ITAG,LNXT,RE,V,RHSR,INSS
     +,N1,LRHSR)
C
C          THIS SUBROUTINE MULTIPLIES THE RIGHT PART OF THE LR
C
C          FACTORIZATION BY THE RHS VECTOR FOR EXTERNAL NODES OF A
```

```
C
C        SUBGRAPH
C
      INTEGER NON(1),LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX RE(1),V(1),RHSR(1),SUM
C
C        NINT      NUMBER OF INTERNAL NODES
C
C        IAD1      ADDRESS IN RHSR WHERE THE INFORMATION ABOUT THE
C                  SUBSTITUTE SUBGRAPH BEGINS
C
C        LRHSR     ADDRESS IN RHSR WHERE THE INFORMATION ABOUT THE
C                  RHS VECTOR STOPS
C
C        INSS      INDICATOR FOR INSUFFICIENT SPACE IN RHSR. WHEN
C                  INSS=1 THE RHS VECTOR FROM ANOTHER RECORD MUST BE
C                  CONSIDERED
C
      INSS=0
      N3=1-N1
      N2=LRHSR-N3
C
C        N1 AND N2 ARE THE LOWEST AND THE HIGHEST INDICES, RESPECTIVELY
C
C        OF NODES FOR WHICH INFORMATION ABOUT THE RHS VECTOR IS STORED
C
C        IN A GIVEN RHS RECORD.
C
      DO 50 JJ=1,NINT
      J=NINT-JJ+1
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 IF(L.LE.0) GOTO 40
      I=ITAG(L)
      IF(I.LE.NINT) GOTO 60
      I=NON(I)
      IF(I.LT.N1) GOTO 60
      IF(I.LE.N2) GOTO 80
      INSS=1
      GOTO 60
   80 SUM=SUM-RE(L)*RHSR(N3+I)
   60 L=LNXT(L)
      GOTO 30
   40 V(K)=SUM
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE RMIRHS(NINT,LCOL,NSEQ,ITAG,LNXT,RE,V)
C
C        THIS SUBROUTINE MULTIPLIES THE RIGHT PART OF THE LR
C
C        FACTORIZATION BY THE RHS VECTOR FOR INTERNAL NODES OF A
C
C        SUBGRAPH
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX RE(1),V(1),SUM
C
C        NINT      NUMBER OF INTERNAL NODES
C
      IF(NINT.LE.1) RETURN
      N1=NINT-1
```

```
      DO 50 JJ=1,N1
      J=NINT-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
   30 IF(L.LE.0) GOTO 40
      I=ITAG(L)
      IF(I.GT.NINT) GOTO 60
      SUM=SUM-RE(L)*V(I)
   60 L=LNXT(L)
      GOTO 30
   40 V(K)=SUM
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CASSOL(IN,IM,LL,SOLR,NINT,INDX,V,NON,INDEX7)
C
C         THIS SUBROUTINE CALCULATES ADDRESSES AND STORES THE SOLUTION
C
C         VECTOR FOR THE INTERNAL NODES OF A SUBGRAPH
C
      INTEGER IN(1),INDX(1),NON(1),IM(1),INDEX7(1)
      COMPLEX SOLR(1),V(1)
C
C         SOLR        SOLUTION RECORD
C
C         NSOLR       INDEX OF SOLUTION RECORD
C
C         IAD1        ADDRESS IN THE SOLUTION RECORD, WHERE THE
C                     INFORMATION ABOUT GIVEN SUBGRAPH BEGINS
C
C         NINT        NUMBER OF EXTERNAL NODES FOR THE SUBGRAPH
C
C         INDX(7)     CURRENT INDEX OF THE SOLUTION RECORD
C
C         LSOL        LENGTH OF SOLUTION RECORD
C
      IM5=IM(5)
      LSOL=4*IM5
      NSEN=LL+1
      NSOLR=NSEN/IM5+1
C
C         STORE OLD SOLUTION RECORD
C
      IF(NSOLR.EQ.INDX(7)) GOTO 9
      IF(NSOLR.GT.IN(15)) GOTO 1
      IF(IN(23).NE.1) CALL WRITMS(7,SOLR,LSOL,INDX(7))
      IF(IN(23).EQ.1) CALL WRITMS(7,SOLR,LSOL,INDX(7),1,1)
      INDX(7)=NSOLR
    9 IAD1=NSEN-(NSOLR-1)*IM5
C
C         CALCULATE THE ADDRESS IN THE SOLUTION RECORD
C
      IAD=IAD1+NINT-1
      IF(IAD.LT.IAD1) RETURN
      IJ=0
      IF(IAD.LE.IM5) GOTO 8
    6 DO 10 I=IAD1,IM5
      IJ=IJ+1
      SOLR(2*I-1)=CMPLX(FLOAT(NON(IJ)),0.)
   10 SOLR(2*I)=V(IJ)
C
C         STORE THE SOLUTION RECORD
```

```
C
      IF(IN(23).NE.1) CALL WRITMS(7,SOLR,LSOL,INDX(7))
      IF(IN(23).EQ.1) CALL WRITMS(7,SOLR,LSOL,INDX(7),1,1)
      INDX(7)=INDX(7)+1
      IAD1=1
      IAD=IAD-IM5
      IF(IAD.GT.IM5) GOTO 6
C
C         WRITE THE SOLUTION VECTOR ASSOCIATED WITH THE SUBGRAPH
C
    8 DO 11 I=IAD1,IAD
      IJ=IJ+1
      SOLR(2*I-1)=CMPLX(FLOAT(NON(IJ)),0.)
   11 SOLR(2*I)=V(IJ)
      LL=LL+NINT
      RETURN
    1 WRITE(6,2)
    2 FORMAT(1H ,/," NUMBER OF SOLUTION RECORDS TOO SMALL",/," INCREASE
     +IN(15)",/)
      IN(3)=1
      RETURN
      END
C
C
      SUBROUTINE CSSLE1 (IN,LCOL,ITAG,CE,DE,V,EPS)
C
C          THIS SUBROUTINE CALLS SIX MAIN SUBROUTINES FOR THE SOLUTION
C
C          OF A LINEAR SYSTEM OF EQUATIONS DEPENDING ON THE PARTICULAR
C
C          JOB
C
      DIMENSION IN(8), LCOL(1), ITAG(1)
      COMPLEX CE(1),DE(1),V(1)
C
C         IN(1)=N       NUMBER OF UNKNOWNS
C
C         IN(2)=LES     MAXIMUM AREA PREDICTED FOR THE SPARSE MATRIX
C
C         IN(3)=IAR     INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND
C                       COLUMN ORDERING OF THE COEFFICIENT MATRIX AND FLAG
C                       FOR INSUFFICIENT AREA
C
C         IN(4)=IAF     FLAG FOR INTERACTIVE SOLUTION
C                       IF IAF=1 FOR SINGLE SOLUTION OR INVERSION OF THE
C                       GIVEN MATRIX THEN RESULTS WILL NOT BE PRINTED OUT
C
C         IN(5)=ISUB1   INDICATOR FOR USER SUBROUTINE; NUMBER OF CHANGES
C                       IN THE COEFFICIENT MATRIX
C
C         IN(6)=ISUB2   INDICATOR FOR USER SUBROUTINE; NUMBER OF INTERNAL
C                       NODES
C
C         IN(7)=NINV    PREDICTED NUMBER OF NONZERO ELEMENTS IN THE
C                       INVERSE MATRIX
C                       IF NINV=1 FOR MULTIPLE NONINTERACTIVE SOLUTION
C                       OR FOR PARTIAL BIFACTORIZATION OF DECOMPPOSED
C                       GRAPH THEN RESULTS WILL NOT BE PRINTED OUT
C
C         IN(8)         INDICATOR FOR KIND OF JOB
C                =1     THE SINGLE SOLUTION SUBROUTINE IS EXECUTED
C                =2     THE MULTIPLE SOLUTION SUBROUTINE IS EXECUTED
C                =3     THE INVERSION SUBROUTINE IS EXECUTED
C                =4     THE SOLUTION IS CALCULATED FOR AN ALTERED COEFFI-
C                       CIENT MATRIX AND/OR RIGHT HAND SIDE VECTOR
```

```
C              =5     THE SOLUTION IS CALCULATED FOR AN ALTERED RIGHT
C                     HAND SIDE VECTOR ONLY
C              =6     BIFACTORIZATION IS PERFORMED FOR INTERNAL
C                     NODES ONLY
C
C        FOR IN(8)=4 THE USER STORES THE NEW COEFFICIENTS IN THE
C        WORKING AREA CE FROM ELEMENT LES*2+1, WHILE ROW AND COLUMN
C        INDICES OF THE NEW COEFFICIENTS ARE STORED IN THE WORKING AREA
C        ITAG FROM ELEMENT 2*LES+1 AND 3*LES+1, RESPECTIVELY. IN(5)
C        STORES THE NUMBER OF NEW COEFFICIENTS.
C
C        FOR IN(8)=4 AND/OR IN(8)=5 THE USER STORES THE NEW RIGHT
C        HAND SIDE ELEMENTS IN THE WORKING AREA V FROM ELEMENT N+1
C
C        DIMENSION OF MATRIX LCOL MUST BE EQUAL TO 4*N+1 FOR INVERSION
C        AND 3*N+1 FOR THE OTHER CASES
C
C        DIMENSION OF MATRIX ITAG MUST BE EQUAL TO 2*LES+NINV FOR
C        IN(8)=1,2,3,5,6 AND 4*LES FOR IN(8)=4
C
C        DIMENSION OF MATRIX CE MUST BE EQUAL TO 2*LES+NINV FOR
C        IN(8)=1,3,5,6 AND 4*LES FOR IN(8)=2,4
C
C        DIMENSION OF MATRIX DE MUST BE EQUAL TO 2*N FOR IN(8)=2,4
C        AND N FOR THE OTHER CASES
C
C        DIMENSION OF MATRIX V MUST BE EQUAL TO 2*N FOR IN(8)=2,4,5
C        AND N FOR THE OTHER CASES
C
      N=IN(1)
      LES=IN(2)
      N1=N+1
      N2=N+2
      N3=N2+N
      N4=N3+N
      LES1=LES+1
      LES2=LES1+LES
      LES3=LES2+LES
      CALL SECOND (TM1)
      IF(IN(8).EQ.6) GO TO 70
      IF(IN(8).EQ.5) GO TO 60
      IF(IN(8).EQ.4) GO TO 50
      IF (IN(8).EQ.3) GO TO 20
      IF (IN(8).EQ.2) GO TO 10
C
C        THE SINGLE SOLUTION SUBROUTINE IS EXECUTED
C
      CALL SSLECN (IN(1),LCOL(1),LCOL(N2),LCOL(N3),ITAG(1),ITAG(LES1),CE
     1(1),CE(LES1),DE(1),V(1),LF,IN(2),IN(3))
      IF (IN(3).EQ.1) RETURN
      IF(IN(4).EQ.1) RETURN
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
      IF(IN(4).NE.1) WRITE(6,120)
120   FORMAT(1H ,/," SOLUTION",/)
      IF (IN(4).NE.1) CALL PRINT (IN(1),V)
      RETURN
C
C        THE MULTIPLE SOLUTION SUBROUTINE IS EXECUTED
C
10    CALL MSLECN (IN(1),LCOL(1),LCOL(N2),LCOL(N3),ITAG(1),ITAG(LES1),CE
     1(1),CE(LES1),DE(1),V(1),LF,IN(2),IN(3),IN(4),V(N1),CE(LES2),CE(LES
     23),DE(N1),IN(5),IN(6),IN(7))
      IF(IN(7).EQ.1)RETURN
```

```
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
      RETURN
C
C         THE INVERSION SUBROUTINE IS EXECUTED
C
20    CALL INVCN (IN(1),LCOL(1),LCOL(N2),LCOL(N3),ITAG(1),ITAG(LES1),CE(
     11),CE(LES1),DE(1),V(1),LF,IN(2),IN(3),EPS,LCOL(N4),ITAG(LES2),CE(L
     2ES2),IN(7))
      IF (IN(3).EQ.1) RETURN
      IF(IN(4).EQ.1) RETURN
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
      IF (IN(4).NE.1) WRITE (6,40)
      IF (IN(4).NE.1) CALL PRINTI (IN(1),CE(LES2),LCOL(N4),ITAG(LES2))
      RETURN
C
C         THE SOLUTION IS CALCULATED FOR AN ALTERED COEFFICIENT
C         MATRIX AND/OR RIGHT HAND SIDE VECTOR
C
50    CALL MSLECC (IN(1),LCOL(1),LCOL(N3),ITAG(1),ITAG(LES1),CE(1),
     1CE(LES1),DE(1),V(1),LF,IN(2),IN(3),IN(4),V(N1),CE(LES2),CE(LES
     23),DE(N1),IN(5),ITAG(LES2),ITAG(LES3))
      IF (IN(3).EQ.1) RETURN
      IF(IN(7).EQ.1)RETURN
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
      IF(IN(7).NE.1) WRITE(6,120)
      IF (IN(7).NE.1) CALL PRINT (IN(1),V)
      RETURN
C
C         THE SOLUTION IS CALCULATED FOR AN ALTERED RIGHT HAND SIDE
C         VECTOR ONLY
C
60    CALL MSLERC (IN(1),LCOL(1),LCOL(N3),ITAG(1),ITAG(LES1),CE(1)
     1,CE(LES1),DE(1),V(1),IN(4),V(N1))
      IF (IN(3).EQ.1) RETURN
      IF(IN(7).EQ.1) RETURN
      CALL SECOND (TM2)
      CPU=TM2-TM1
      WRITE (6,30) CPU
      IF(IN(7).NE.1) WRITE(6,120)
      IF (IN(7).NE.1) CALL PRINT (IN(1),V)
      RETURN
C
C
C         BIFACTORIZATION IS PERFORMED FOR INTERNAL NODES ONLY
C
70    CALL PABIDES(IN(1),LCOL(1),LCOL(N2),LCOL(N3),ITAG(1),ITAG(LES1),
     1CE(1),CE(LES1),DE(1),LF,IN(2),IN(3),IN(6))
      IF(IN(3).EQ.1)RETURN
      IF(IN(7).NE.2)RETURN
      CALL SECOND(TM2)
      CPU=TM2-TM1
      WRITE(6,30)CPU
      IF(IN(1).EQ.1.AND.LCOL(1).EQ.0) GOTO 90
      WRITE(6,110)
      CALL PRINTS (IN(1),CE(1),LCOL(1),ITAG(1),ITAG(LES1))
      WRITE(6,100)
      CALL PRINTS(IN(1),CE(LES1),LCOL(1),ITAG(1),ITAG(LES1))
   90 WRITE(6,130)
      CALL PRINT(IN(1),DE)
```

```
130   FORMAT( 1H ,/," DIAGONAL ELEMENTS",/)
100   FORMAT( 1H ,/," UPPER TRIANGULAR PART ",/)
110   FORMAT( 1H ,/," LOWER TRIANGULAR PART ",/)
80    RETURN
30    FORMAT (1H ,/," CPU TIME:",F8.3," SECONDS",/)
40    FORMAT (1H ,/," INVERSE MATRIX",/)
      END
C
C
      SUBROUTINE SSLECN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IA
     1R)
C
C        THIS SUBROUTINE SOLVES A SYSTEM OF SPARSE LINEAR EQUATIONS
C
C        WHOSE COEFFICIENT MATRIX IS SPARSE AND NONSYMMETRICAL
C
C        BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C        " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C        - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C        MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C        J.K.REID, EDITOR, ACADEMIC PRESS, 1971, PP.75-96
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      COMPLEX CE(1),V(1),RE(1),DE(1)
C
C        N     IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C        LCOL STARTING POSITION OF COLUMNS
C             N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C             POSITION. DIMENSION IS EQUAL TO N+1
C
C        NOZE NUMBER OF NONZERO TERMS
C             CALCULATED BY THE DATBCN SUBROUTINE
C             DIMENSION IS EQUAL TO N
C
C        NSEQ SEQUENCE OF PIVOTAL INDICES
C             CALCULATED BY THE DATACN SUBROUTINE
C             DIMENSION IS EQUAL TO N
C
C        ITAG ROW INDEX OF ELEMENTS STORED IN CE
C             DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C             ELEMENTS
C
C        LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C             DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C             ELEMENTS
C             CALCULATED BY THE DATACN SUBROUTINE
C
C        CE   IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
C
C        RE   AN ARRAY HAVING ROWWISE STORED TERMS OF THE COEFFICIENT
C             MATRIX. CALCULATED BY THE DATA SUBROUTINE
C
C        DE   AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF CO-
C             EFFICIENT MATRIX. DATA REQUIRED FROM THE USER
C
C        V    AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C             SIDE VECTOR
C             AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C        LF   INDICATOR FOR THE NEXT VACANT LOCATION
```

```
C
C      LES   NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C      IAR   INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C            ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C            AREA
C
      CALL DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES)
C
      CALL DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      NINT=N
      CALL SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR,NINT)
      IF (IAR.EQ.1) RETURN
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
      IF (IAR.EQ.1) RETURN
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      RETURN
      END
C
C
      SUBROUTINE MSLECN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IA
     1R,IAF,VV,CCE,RRE,DDE,ISUB1,ISUB2,NINV)
C
C      THIS SUBROUTINE OBTAINS MULTIPLE SOLUTIONS OF A SYSTEM OF
C
C      LINEAR EQUATIONS WHOSE COEFFICIENT MATRIX IS SPARSE AND
C
C      NONSYMMETRICAL
C
C      AN INTERACTIVE OR NONINTERACTIVE VERSION CAN BE USED TO HANDLE
C
C      DIFFERENT CHANGES IN BOTH COEFFICIENT MATRIX AND RIGHT HAND
C
C      SIDE VECTOR
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      COMPLEX CE(1),V(1),RE(1),DE(1),VV(1),CCE(1),RRE(1),DDE(1)
C
C      N     IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C      LCOL  STARTING POSITION OF COLUMNS
C            N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C            POSITION
C            DIMENSION IS EQUAL TO N+1
C
C      NOZE  NUMBER OF NONZERO TERMS
C            CALCULATED BY THE DATBCN SUBROUTINE
C            DIMENSION IS EQUAL TO N
C
C      NSEQ  SEQUENCE OF PIVOTAL INDICES
C            CALCULATED BY THE DATACN SUBROUTINE
C            DIMENSION IS EQUAL TO N
C
C      ITAG  ROW INDEX OF ELEMENTS STORED IN CE
C            DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C            ELEMENTS
C
C      LNXT  LOCATION OF THE NEXT TERM IN EACH COLUMN
C            DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C            ELEMENTS
C            CALCULATED BY THE DATACN SUBROUTINE
C
C      CE    IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
C
```

```
C        RE    AN ARRAY HAVING ROWWISE STORED TERMS OF THE COEFFICIENT
C              MATRIX CALCULATED BY THE DATA SUBROUTINE
C
C        DE    AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF CO-
C              EFFICIENT MATRIX - DATA REQUIRED FROM THE USER
C
C        V     AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C              SIDE VECTOR
C              AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C        LF    INDICATOR FOR THE NEXT VACANT LOCATION
C
C        LES   NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C        IAR   INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C              ORDERING OF THE COEFFICIENT MATRIX AND FLAG FOR INSUFFI-
C              CIENT AREA
C
C        IAF   FLAG FOR INTERACTIVE SOLUTION. FOR IAF ODD THE SOLUTION
C              IS OBTAINED DURING AN INTERACTIVE SESSION, WHILE FOR
C              IAF EVEN THE SOLUTION IS GENERATED ON THE BASIS OF DATA
C              STORED BY THE USER
C              FOR IAF EQUAL:
C
C              0 OR 1 ONLY THE SOLUTION OF THE GIVEN SYSTEM IS
C                     CALCULATED
C              2 OR 3 THE SOLUTION OF THE GIVEN AND ADJOINT SYSTEM MAY
C                     BE CALCULATED
C              4 OR 5 ONLY THE SOLUTION OF THE ADJOINT SYSTEM IS CALCULA
C              TED
C
C        VV,CCE,RRE,DDE -ADDITONAL MATRICES STORING INTERMEDIATE
C              RESULTS FOR THE INTERACTIVE SOLUTION
C
C        ISUB1,ISUB2 -INDICATORS FOR USER SUBROUTINES USUB1,USUB2.
C              IF EQUAL 1 USER SUBROUTINES WILL BE CALLED BEFORE
C              ALTERING THE COEFFICIENT MATRIX AND/OR RIGHT HAND
C              SIDE VECTOR
C
C        FILE NO 6 IS NECESSARY FOR PRINTING THE RESULTS
C
      DO 10 I=1,N
      VV(I)=V(I)
10    CONTINUE
      CALL DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES)
C
      CALL DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      NINT=N
      CALL SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR,NINT)
      IF (IAR.EQ.1) RETURN
      DO 20 I=1,LES
      CCE(I)=CE(I)
      RRE(I)=RE(I)
20    CONTINUE
      DO 30 I=1,N
      DDE(I)=DE(I)
30    CONTINUE
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
      IF (IAR.EQ.1) RETURN
      IF (IAF.GT.3) GO TO 50
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      WRITE (6,70)
      IF(NINV.NE.1) WRITE(6,100)
100   FORMAT(1H ,/," SOLUTION",/)
```

```
      IF(NINV.NE.1)CALL PRINT(N,V)
      DO 40 I=1,N
      V(I)=VV(I)
40    CONTINUE
50    IF (IAF.LT.2) GO TO 60
      CALL ASLECN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      WRITE (6,80)
      IF(NINV.NE.1) WRITE(6,100)
      IF(NINV.NE.1) CALL PRINT(N,V)
60    CONTINUE
      IF(IAF.EQ.0.OR.IAF.EQ.2.OR.IAF.EQ.4) GO TO 90
      CALL RSLECN (N,IAF,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,VV,LF,LES,I
     1AR,CCE,RRE,DDE,ISUB1,ISUB2)
C
90    DO 310 I=1,LES
      CE(I)=CCE(I)
      RE(I)=RRE(I)
310   CONTINUE
      DO 320 I=1,N
      DE(I)=DDE(I)
320   CONTINUE
      RETURN
C
C
70    FORMAT (1H ,/," END OF FIRST SOLUTION")
80    FORMAT (1H ,/," SOLUTION OF ADJOINT SYSTEM")
      END
C
C
      SUBROUTINE INVCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,IAR
     1,EPS,LCLI,ITGI,INV,NINV)
C
C         THIS SUBROUTINE CALCULATES THE INVERSION OF THE GIVEN SPARSE
C
C         MATRIX. IF AN ELEMENT OF THE INVERSE MATRIX HAS AN ABSOLUTE
C
C         VALUE SMALLER THAN EPS THEN A ZERO VALUE IS ASSUMED. THE FINAL
C
C         INVERTED MATRIX IS STORED IN SPARSE FORM SIMILAR TO THE INPUT
C
C         DATA
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1)
      INTEGER LCLI(1),ITGI(1)
      COMPLEX CE(1),V(1),RE(1),DE(1)
      COMPLEX INV(1)
C
C         N    IS THE NUMBER OF UNKNOWNS,ORDER OF THE MATRIX
C
C         LCOL STARTING POSITION OF COLUMNS
C              N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT
C              POSITION.
C              DIMENSION IS EQUAL TO N+1
C
C         NOZE NUMBER OF NONZERO TERMS
C              CALCULATED BY THE DATBCN SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         NSEQ SEQUENCE OF PIVOTAL INDICES
C              CALCULATED BY THE DATACN SUBROUTINE
C              DIMENSION IS EQUAL TO N
C
C         ITAG ROW INDEX OF ELEMENTS STORED IN CE
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
```

```
C
C       LNXT LOCATION OF THE NEXT TERM IN EACH COLUMN
C            DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C            ELEMENTS
C            CALCULATED BY THE DATACN SUBROUTINE
C
C       CE   IS THE MATRIX OF COMPLEX COEFFICIENTS OF A GIVEN EQUATION
C
C       RE   AN ARRAY HAVING ROWWISE STORED TERMS OF THE COEFFICIENT
C            MATRIX. CALCULATED BY THE DATA SUBROUTINE
C
C       DE   AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF CO-
C            EFFICIENT MATRIX. DATA REQUIRED FROM THE USER
C
C       V    AN  ARRAY OF DIMENSION N STORING THE RIGHT HAND
C            SIDE VECTOR
C            AT THE OUTPUT V CONTAINS THE SOLUTION VECTOR
C
C       LF   INDICATOR FOR THE NEXT VACANT LOCATION
C
C       LES  NUMBER OF ELEMENTS IN MATRICES ITAG,LNXT,RE & CE
C
C       IAR  INDICATOR FOR CHECKING STRUCTURAL SYMMETRY AND COLUMN
C            ORDERING OF COEFFICIENT MATRIX AND FLAG FOR INSUFFICIENT
C            AREA
C
C       EPS  ALL ELEMENTS OF THE INVERSE MATRIX HAVING ABSOLUTE VALUE
C            LESS THAN EPS WILL BE ASSUMED ZERO VALUE AND WILL NOT
C            BE STORED
C
C       LCLI STARTING POSITON OF COLUMNS IN THE INVERSE MATRIX
C            N+1-ST ELEMENT OF LCLI INDICATES THE FIRST VACANT POSITON
C            DIMENSION IS EQUAL TO N+1
C
C       ITGI ROW INDEX OF ELEMENTS STORED IN INV
C            DIMENSION IS NOT LESS THAN NINV
C
C       INV  COLUMNWISE STORED INVERSE MATRIX
C            DIMENSION IS NOT LESS THAN NINV
C
C       NINV NUMBER OF NONZERO ELEMENTS IN INV
C
      CALL DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES)
C
      CALL DATBCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      NINT=N
      CALL SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR,NINT)
      IF (IAR.EQ.1) RETURN
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
      IF (IAR.EQ.1) RETURN
C
C       PREPARE THE RIGHT HAND SIDE VECTOR
C
      LICZ=0
      LCLI(1)=1
      DO 30 I=1,N
      DO 10 J=1,N
      V(J)=CMPLX(0.,0.)
10    CONTINUE
      V(I)=CMPLX(1.,0.)
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      DO 20 J=1,N
      IF (CABS(V(J)).LE.EPS) GO TO 20
      LICZ=LICZ+1
```

```
C
C         NUMBER OF ELEMENTS IN THE INVERSE IS GREATER THAN DECLARED
C
      IF (LICZ.GT.NINV) GO TO 40
      ITGI(LICZ)=J
      INV(LICZ)=V(J)
20    CONTINUE
      LCLI(I+1)=LICZ+1
30    CONTINUE
      RETURN
40    NINV=0
      WRITE (6,50)
      RETURN
C
50    FORMAT ("  NUMBER OF ELEMENTS IN THE INVERSE IS GREATER THAN DECLA
     1RED")
      END
C
C
      SUBROUTINE MSLECC(N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V,LF,LES,
     +IAR,IAF,VV,CCE,RRE,DDE,NCHM,IRN,ICN)
C
C         THIS SUBROUTINE REPEATS THE SOLUTION OF THE LINEAR SPARSE
C
C         SYSTEM FOR ALTERED COEFFICIENTS AND/OR RIGHT HAND SIDE VECTOR
C
C         IAF INTERACTIVE VECTOR SHOULD BE EQUAL 0,2 OR 4 DEPENDING ON
C
C         THE DESIRED TYPE OF ANALYSIS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1),IRN(1),ICN(1)
      COMPLEX COEFF,CE(1),RE(1),DE(1),V(1),VV(1),CCE(1),RRE(1),DDE(1)
C
C         NEW COEFFICIENT SHOULD BE PLACED IN CCE AND ITS ROW AND
C
C         COLUMN NUMBERS IN IRN AND ICN RESPECTIVELY. NCHM IS THE
C
C         NUMBER OF NEW COEFFICIENTS. NEW RIGHT HAND SIDE ELEMENTS
C
C         SHOULD BE PLACED IN VV.
C
      IF (NCHM.EQ.0) GO TO 360
      DO 330 I=1,NCHM
      NROW=IRN(I)
      NCOL=ICN(I)
      COEFF=CCE(I)
      IF(NROW.GT.N.OR.NROW.LE.0.OR.NCOL.GT.N.OR.NCOL.LE.0) GO TO 100
C
C         THE NEW ELEMENT MUST BE ENTERED INTO CE OR RE DEPENDING ON
C         WHICH OF THESE MATRICES IS TO STORE THE VALUE OF THIS ELEMENT
C         AFTER THE SIMULATION AND ORDERING PROCESS
C
      CALL CHNGE (IAR,NROW,NCOL,COEFF,LCOL,ITAG,LNXT,CE,RE,DE)
      IF(IAR.EQ.1) RETURN
330   CONTINUE
C
C         REPEAT BIFACTORIZATION
C
      IAR=0
C
C         STORE THE NEW VALUES OF THE ELEMENTS OF THE COEFFICIENT
C         MATRICES
C
      DO 340 I=1,LES
      CCE(I)=CE(I)
```

```
      RRE(I)=RE(I)
340   CONTINUE
      DO 350 I=1,N
      DDE(I)=DE(I)
350   CONTINUE
      NINT=N
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
C
C          SOLUTION WITH NEW RIGHT HAND SIDE VECTOR
C
  360 CONTINUE
      DO 380 J=1,N
      V(J)=VV(J)
380   CONTINUE
      IF (IAF.EQ.0) GO TO 420
      CALL ASLECN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      DO 410 J=1,N
      V(J)=VV(J)
410   CONTINUE
420   IF (IAF.EQ.4) GO TO 430
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
430   CONTINUE
      RETURN
  100 WRITE (6,490)
 490  FORMAT(1H ," ELEMENT OUT OF THE MATRIX AREA",/)
      RETURN
      END
C
C
      SUBROUTINE MSLERC(N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V,IAF,VV)
C
C          THIS SUBROUTINE REPEATS THE SOLUTION OF THE LINEAR SPARSE
C
C          SYSTEM FOR ALTERED RIGHT HAND SIDE VECTOR
C
C          IAF INTERACTIVE VECTOR SHOULD BE EQUAL 0,2 OR 4 DEPENDING ON
C
C          THE DESIRED TYPE OF ANALYSIS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),V(1),VV(1)
C
C          NEW RIGHT HAND SIDE ELEMENTS SHOULD BE PLACED IN VV
C
      DO 380 J=1,N
      V(J)=VV(J)
380   CONTINUE
      IF (IAF.EQ.0) GO TO 420
      CALL ASLECN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      DO 410 J=1,N
      V(J)=VV(J)
410   CONTINUE
420   IF (IAF.EQ.4) GO TO 430
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
430   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE PABIDES(IN,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,LF,LES,IA
     +R,NINT)
C
C          THIS SUBROUTINE PERFORMS PARTIAL BIFACTORIZATION OF A
C
C          DECOMPOSED SYSTEM OF LINEAR EQUATIONS
```

```
C
      INTEGER ITAG(1),LCOL(1),LNXT(1),NOZE(1),NSEQ(1),IN(1)
      COMPLEX CE(1),RE(1),DE(1)
C
C         NINT IS THE NUMBER OF INTERNAL NODES
C
      N=IN(1)
      CALL DATACN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES)
      CALL DATBCN(IN,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
      IAR=0
      CALL SORDCN(N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR,NINT)
      IF(IAR.EQ.1) RETURN
C
C         STORE UPPER AND LOWER PARTS
C
      LES2=LES*2
      DO 1 I=1,LES2
    1 CE(LES2+I)=CE(I)
C
C         STORE DIAGONAL ELEMENTS
C
      DO 2 I=1,N
2     DE(N+I)=DE(I)
C
      CALL REDUCN(N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
      RETURN
      END
C
C
      SUBROUTINE DATAC (N,NEL,AK,NCOL,NROW,LCOL,ITAG,CE,DE,LES,IAR)
C
C         THIS ROUTINE CHANGES THE DATA STORED IN THE COEFFICIENT MATRIX
C
C         AK TO THE FORM DESIRED BY THE SUBROUTINES SOLVING THE SPARSE
C
C         SYSTEMS.
C
C         COLUMN NUMBERS OF SUCCESSIVE ELEMENTS FROM AK ARE STORED IN
C
C         MATRIX NCOL, WHILE ROW NUMBERS IN NROW.
C
C         THE PROGRAM GENERATES MATRICES LCOL, ITAG, CE, DE, WHERE
C
C         N    NUMBER OF EQUATIONS
C
C         NEL  NUMBER OF ELEMENTS IN AK
C
C         LCOL STARTING POSITIONS OF COLUMNS
C              N+1-ST ELEMENT OF LCOL INDICATES THE FIRST VACANT POSITON
C              DIMENSION IS EQUALTO N+1
C
C         ITAG ROW INDEX FOR ELEMENTS STORED IN CE
C              DIMENSION IS NOT SMALLER THAN THE NUMBER OF NONZERO
C              ELEMENTS
C              CALCULATED BY THE DATACN SUBROUTINE
C
C         CE   COLUMNWISE STORED MATRIX OF NONZERO COEFFICIENTS
C              OF A GIVEN EQUATION
C
C         DE   AN ARRAY OF DIMENSION N STORING DIAGONAL TERMS OF THE
C              COEFFICIENT MATRIX
C
C         LES  PREDICTED MAXIMUM AREA USED BY MATRIX CE
C
C         IAR  FLAG FOR INSUFFICIENT AREA
```

```
C
      COMPLEX AK(1),CE(1),DE(1)
      INTEGER NCOL(1),NROW(1),LCOL(1),ITAG(1)
      N1=N+1
      DO 10 I=1,N1
      LCOL(I)=0
10    CONTINUE
      LCOL(1)=1
      DO 20 I=1,NEL
      IF (NCOL(I).EQ.NROW(I)) GO TO 20
      LCOL(NCOL(I)+1)=LCOL(NCOL(I)+1)+1
20    CONTINUE
      DO 40 I=1,N
      LCOL(I+1)=LCOL(I)+LCOL(I+1)
40    CONTINUE
C
C         CALCULATE MATRICES DE,CE,ITAG
C
      DO 50 I=1,NEL
      NRWI=NROW(I)
      NCLI=NCOL(I)
      IF(NCLI.EQ.NRWI) GOTO 70
      LCNC=LCOL(NCLI)
      CE(LCNC)=AK(I)
      ITAG(LCNC)=NRWI
      LCOL(NCLI)=LCNC+1
      GOTO 50
   70 DE(NCLI)=AK(I)
50    CONTINUE
C
C         CALCULATE MATRIX LCOL
C
      DO 60 I=1,N
      J=N-I+2
      LCOL(J)=LCOL(J-1)
60    CONTINUE
      LCOL(1)=1
      DO 80 I=1,N
   80 IF(LCOL(I).EQ.LCOL(I+1)) LCOL(I)=0
      RETURN
      END
C
C
      SUBROUTINE DATACN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NSEQ,LNXT,LF
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1)
      LF=LCOL(N+1)
      DO 10 I=1,LES
      LNXT(I)=I+1
10    CONTINUE
      LNXT(LES)=0
      DO 20 I=LF,LES
      CE(I)=CMPLX(0.,0.)
20    CONTINUE
C
C         CALCULATE MATRICES NSEQ & LNXT
C
      DO 50 I=1,N
      J=I+1
      IF (LCOL(I).EQ.0) GO TO 50
```

```
30    IF (LCOL(J).NE.0) GO TO 40
      J=J+1
      GO TO 30
40    LNXT(LCOL(J)-1)=0
50    NSEQ(I)=I
      RETURN
      END
C
C
      SUBROUTINE DATBCN (IN,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,LES,IAR)
C
C         THIS SUBROUTINE CALCULATES VALUES OF THE FOLLOWING VARIABLES:
C
C         NOZE,RE
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1),IN(1)
      COMPLEX CE(1),RE(1)
C
C         IF IAR.EQ.2 CHECK SYMMETRY
C
      N=IN(1)
      IF (IAR.EQ.2) CALL SYMCN (IN,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
      IF(ITAG(NE).LE.0) GOTO 20
      NOZE(I)=NOZE(I)+1
      NE=LNXT(NE)
      GO TO 10
20    CONTINUE
      DO 30 I=1,LES
      RE(I)=CMPLX(0.,0.)
30    CONTINUE
C
C         CALCULATE ROWWISE STORED COEFFICIENT MATRIX
C
      DO 70 I=1,N
      NE=LCOL(I)
40    IF (NE.EQ.0) GO TO 70
      IF(ITAG(NE).LE.0) GOTO 70
      NR=ITAG(NE)
      LI=LCOL(NR)
50    IF (ITAG(LI).EQ.I) GO TO 60
      LI=LNXT(LI)
      GO TO 50
60    RE(LI)=CE(NE)
      NE=LNXT(NE)
      GO TO 40
70    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SYMCN (IN,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,LF)
C
C         THIS SUBROUTINE CHECKS THE STRUCTURAL SYMMETRY AND ORDERING OF
C
C         COLUMNS OF THE COEFFICIENT MATRIX. IF THE MATRIX IS NOT
C
C         SYMMETRICAL, NEW ZERO COEFFICIENTS ARE ADDED TO REACH SYMMETRY
C
```

```
C         COLUMNS ARE REORDERED IF NECESSARY
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1),IN(1)
      COMPLEX CE(1)
      N=IN(1)
      IS=0
      DO 70 I=1,N
      LC=LCOL(I)
      IF (LC.EQ.0) GO TO 70
10    IR=ITAG(LC)
      IF(IR.EQ.0) GOTO 70
C
C         CHECK IF THERE EXISTS AN ELEMENT SYMMETRICAL TO THE ONE
C         CONSIDERED
C
      NLC=LCOL(IR)
20    IF (NLC.EQ.0) GO TO 30
      LLC=NLC
      LLR=ITAG(LLC)
      IF (LLR.EQ.I) GO TO 60
      NLC=LNXT(LLC)
      GO TO 20
C
C         MATRIX IS NONSYMMETRICAL-ADD NECESSARY ZERO COEFFICIENT
C
30    IF (LCOL(IR).NE.0) GO TO 40
      LCOL(IR)=LF
      GO TO 50
40    LNXT(LLC)=LF
50    CE(LF)=CMPLX(0.,0.)
      IS=1
      ITAG(LF)=I
      NOZE(IR)=NOZE(IR)+1
      IF=LF
      LF=LNXT(IF)
      LNXT(IF)=0
C
C         PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C         CONSIDERATION
C
60    LC=LNXT(LC)
      IF (LC.NE.0) GO TO 10
70    CONTINUE
      IF (IS.EQ.1.AND.(IN(7).EQ.0.OR.IN(7).EQ.2)) WRITE (6,130)
C
C         ROWS AND COLUMNS ARE ORDERED SUCH THAT INCREASING VALUES OF
C
C         INDICES DESCRIBE THE NEXT NONZERO ELEMENTS IN EACH ROW AND
C
C         COLUMN
C
      IS=0
      DO 120 I=1,N
      I1=LCOL(I)
80    II=I1
90    I2=LNXT(II)
      IF (I2.EQ.0) GO TO 110
      IF (ITAG(I1).LT.ITAG(I2)) II=I2
      IF (ITAG(I1).LT.ITAG(I2)) GO TO 90
      IS=1
      IF (I1.EQ.LCOL(I)) GO TO 100
C
C         EXCHANGE TWO ELEMENTS
C
      LNXT(I0)=I2
```

```
      LNXT(I1)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
100   LCOL(I)=I2
      LNXT(I1)=LNXT(I2)
      LNXT(I2)=I1
      I1=I2
      GO TO 90
C
C         PROCEED WITH THE NEXT ELEMENT IN THE COLUMN UNDER
C         CONSIDERATION
C
110   I0=I1
      I1=LNXT(I1)
      IF (I1.EQ.0) GO TO 120
      GO TO 80
120   CONTINUE
      IF(IN(7).EQ.1.OR.IN(7).EQ.3)RETURN
      IF (IS.EQ.1) WRITE (6,140)
      RETURN
C
130   FORMAT (1H ,/," INITIAL COEFFICIENT MATRIX NONSYMMETRICAL")
140   FORMAT (1H ,/," REORDERING OF THE ELEMENTS OF EACH COLUMN")
      END
C
C
      SUBROUTINE SORDCN (N,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,LF,IAR,
     +NINT)
C
C         THIS SUBROUTINE SIMULATES AND ORDERS GAUSS ELIMINATION
C
C         FOR NONSYMMETRICAL MATRICES
C
C         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971, PP.75-96
C
      INTEGER LCOL(1),NOZE(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1)
      N1=N-1
      IF(N1.EQ.0) RETURN
      DO 140 J=1,N1
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      J1=J+1
      IF(J1.GT.NINT) GO TO 160
C
C         FIND THE COLUMN M WITH THE MINIMUM NUMBER OF ELEMENTS
C
      DO 10 I=J1,NINT
      K=NSEQ(I)
      IF (NOZE(K).GE.MIN) GO TO 10
      MIN=NOZE(K)
      M=I
10    CONTINUE
C
C         INTERCHANGE INDICES WITHIN NSEQ FOR CURRENT J AND M
```

```
C
  160 CONTINUE
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
20    IF (LK.LE.0) GO TO 140
C
C          TAKE THE NEXT ELEMENT FROM MINIMAL COLUMN M
C
      K=ITAG(LK)
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
      I=ITAG(L)
30    IF (I-IP) 50,40,90
C
C          TWO NONZERO ENTRIES ARE SUMMED
C
40    LA=L
      L=LNXT(L)
      I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 100
50    IF (I.NE.KP) GO TO 60
C
C          DELETE UPPER (OR LOWER) DIAGONAL ENTRY REMAINING AFTER
C          PERFORMING GAUSS ELIMINATION
C
      LN=LNXT(L)
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LNXT(L)=LF
      LF=L
      CE(L)=CMPLX(0.,0.)
      RE(L)=CMPLX(0.,0.)
      NOZE(K)=NOZE(K)-1
      L=LN
      GO TO 70
60    LA=L
      L=LNXT(L)
70    IF (L.LE.0) GO TO 80
      I=ITAG(L)
      GO TO 30
C
C          LAST ELEMENT IN THE COLUMN
C
80    IF (LI.LE.0) GO TO 120
      I=N+1
      GO TO 30
90    IF (IP.EQ.K) GO TO 100
      IF (LF.LE.0) GO TO 130
C
C          ADD NEW NONZERO ENTRY
C
      LN=LF
      IF (LA.GT.0) LNXT(LA)=LN
      IF (LA.LE.0) LCOL(K)=LN
      LF=LNXT(LN)
      LNXT(LN)=L
      ITAG(LN)=IP
      NOZE(K)=NOZE(K)+1
      LA=LN
C
```

```
C         TAKE THE NEXT ELEMENT FROM THE ROW WHOSE INDEX IS THE SAME AS
C
C         THE COLUMN INDEX FOR WHICH THE GAUSS ELIMINATION PROCESS IS
C
C         EXECUTED
C
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
C
C         TAKE THE NEXT ELEMENT FROM THE COLUMN FOR WHICH THE GAUSS
C         ELIMINATION PROCESS IS EXECUTED
C
120   LK=LNXT(LK)
      GO TO 20
130   WRITE (6,150)
      IAR=1
      RETURN
140   CONTINUE
      RETURN
C
150   FORMAT (" DIMENSIONED AREA TO SMALL IN SORDCN")
      END
C
C
      SUBROUTINE REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
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         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971, PP.75÷96
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),D,CF,RF
C
C         THE ELEMENTS OF FACTOR MATRICES ARE CALCULATED IN EACH
C
C         STEP ACCORDING TO THE FORMULAS:
C
C         L(II)=1/A(II)
C
C         L(IJ)=A(IJ)
C
C         R(IK)=A(IK)/A(II), FOR J,K≠I AND A() BEING THE COEFFICIENT
C
C         MATRIX.
C
      DO 100 J=1,NINT
      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)
      IF(K.LE.0) RETURN
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    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)
      I=ITAG(L)
      IF (L.LE.0) I=N+1
      GO TO 80
70    IF (IP.NE.K) GO TO 80
      DE(K)=DE(K)-CF*CE(LI)
80    LI=LNXT(LI)
      IF (LI.LE.0) GO TO 90
      IP=ITAG(LI)
      GO TO 40
90    LK=LNXT(LK)
      IF (LK.GT.0) GO TO 20
100   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
C
C         THIS SUBROUTINE SOLVES LINEAR EQUATIONS BY A SEQUENCE OF
C
C         MATRIX MULTIPLICATIONS
C
C         BASED UPON THE ZOLLENKOPF ALGORITHM PRESENTED IN
C
C         " LARGE SPARSE SETS OF LINEAR EQUATIONS "
C
C         - PROCEEDINGS OF THE OXFORD CONFERENCE OF THE INSTITUTE OF
C
C         MATHEMATICS AND ITS APPLICATIONS, HELD IN APRIL 1970-
C
C         J.K.REID, EDITOR, ACADEMIC PRESS, 1971, PP.75÷96
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),V(1),CF,SUM
C
C         FIRST PART OF THE SOLUTION
C         B1=L(N)*L(N-1)*...*L(1)*B.
C
      IF(N.EQ.1) GOTO 60
      DO 20 J=1,N
```

```
      K=NSEQ(J)
      CF=DE(K)*V(K)
      V(K)=CF
      L=LCOL(K)
 10   IF (L.LE.0) GO TO 20
      I=ITAG(L)
      V(I)=V(I)-CE(L)*CF
      L=LNXT(L)
      GO TO 10
 20   CONTINUE
C
C         SECOND PART OF THE SOLUTION
C         X=R(1)*R(2)*...*R(N)*B1
C
      N1=N-1
      DO 50 JJ=1,N1
      J=N-JJ
      K=NSEQ(J)
      SUM=V(K)
      L=LCOL(K)
 30   IF (L.LE.0) GO TO 40
      I=ITAG(L)
      SUM=SUM-RE(L)*V(I)
      L=LNXT(L)
      GO TO 30
 40   V(K)=SUM
 50   CONTINUE
      RETURN
   60 V(1)=V(1)*DE(1)
      RETURN
      END
C
C
      SUBROUTINE ASLECN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
C
C         THIS SUBROUTINE SOLVES THE LINEAR EQUATIONS OF THE ADJOINT
C
C         SYSTEM BY A SEQUENCE OF MATRIX MULTIPLICATIONS
C
      INTEGER LCOL(1),NSEQ(1),ITAG(1),LNXT(1)
      COMPLEX CE(1),RE(1),DE(1),V(1),CF,SUM
C
C         FIRST PART OF THE SOLUTION
C         B1=R(N)*R(N-1)*...*R(1)*B
C
      N1=N-1
      DO 20 J=1,N1
      K=NSEQ(J)
      CF=V(K)
      L=LCOL(K)
 10   IF (L.LE.0) GO TO 20
      I=ITAG(L)
      V(I)=V(I)-RE(L)*CF
      L=LNXT(L)
      GO TO 10
 20   CONTINUE
C
C         SECOND PART OF THE SOLUTION
C         X=L(1)*L(2)*...*L(N)*B1
C
      N2=N+1
      DO 50 JJ=1,N
      J=N2-JJ
      K=NSEQ(J)
      SUM=V(K)*DE(K)
```

```
      L=LCOL(K)
30    IF (L.LE.0) GO TO 40
      I=ITAG(L)
      SUM=SUM-CE(L)*V(I)*DE(K)
      L=LNXT(L)
      GO TO 30
40    V(K)=SUM
50    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE RSLECN (N,IAF,LCOL,NOZE,NSEQ,ITAG,LNXT,CE,RE,DE,V,VV,LF
     1,LES,IAR,CCE,RRE,DDE,ISUB1,ISUB2)
C
C          THIS SUBROUTINE REPEATS THE SOLUTION OF THE LINEAR SPARSE
C
C          SYSTEM FOR ALTERED COEFFICIENTS AND/OR RIGHT HAND SIDE
C
C          VECTOR
C
C          IAF   INTERACTIVE FACTOR SHOULD BE ODD (1,3 OR 5)  IF CHANGES
C
C          WILL BE DONE IN THE INTERACTIVE MODE
C
      INTEGER LCOL(1),ITAG(1),LNXT(1),NOZE(1),NSEQ(1)
      COMPLEX COEFF,CE(1),DE(1),V(1),RE(1),VV(1),CCE(1),RRE(1),DDE(1)
      REAL TEXT(80)
C
C          LM,LE COUNTERS OF CHANGES IN COEFFICIENT N+1MATRIX AND RIGHT H
C                SIDE VECTOR, RESPECTIVELY
C
C          IEND  IS USED TO STOP THE INTERACTIVE PROCESS
C
      LM=0
      LE=0
      IADR=1
10    IEND=0
C
C          RESET THE VALUES OF THE ELEMENTS OF THE COEFFICIENT MATRIX
C          TO THOSE OF THE CURRENT SET IN PREPARATION FOR NEW CHANGES
C
      DO 20 I=1,LES
      CE(I)=CCE(I)
      RE(I)=RRE(I)
20    CONTINUE
      DO 30 I=1,N
      DE(I)=DDE(I)
30    CONTINUE
C
C          USER SUBROUTINE
C          THE USER HAS ACCESS TO THE CURRENT FORM OF THE ANALYZED SYSTEM
C          MATRIX V CONTAINS THE CURRENT RIGHT HAND SIDE VECTOR
C          CE,DE,RE CONTAIN THE CURRENT COEFFICIENT MATRIX
C
      IF (ISUB1.LE.0) GO TO 50
      CALL USUB1 (ISUB1,LCOL,ITAG,LNXT,CE,RE,DE,V)
      IEND=1
      DO 40 I=1,N
      VV(I)=V(I)
40    CONTINUE
50    CONTINUE
      WRITE (6,440)
      CALL READL (TEXT)
      CALL FINDSYM (TEXT,1HN,LCHMD
```

```
      CALL FINDSYM (TEXT,1H*,NOQNS)
      IF (NOQNS.EQ.1) IADR=1
      IF (LCHM.EQ.1.AND.ISUB1.LE.0) GO TO 160
      ISUB1=ISUB1-1
      IF (LCHM.EQ.1) GO TO 130
60    WRITE (6,450)
      READ*,NCHM
      IF (NCHM.GT.0) GO TO 70
      WRITE (6,460)
      GO TO 60
70    IEND=1
      LM=LM+1
C
C          CHANGE SUCCESSIVELY ALL MATRIX ELEMENTS YOU WANT
C
      DO 120 I=1,NCHM
80    WRITE (6,470)
90    CONTINUE
      READ*,NROW,NCOL
      IF (NROW.GT.N.OR.NROW.LE.0.OR.NCOL.GT.N.OR.NCOL.LE.0) GO TO 100
      WRITE (6,480)
      READ*,COEFF
C
C          THE NEW ELEMENT MUST BE ENTERED INTO CE OR RE DEPENDING ON
C          WHICH OF THESE MATRICES IS TO STORE THE VALUE OF THIS ELEMENT
C          AFTER THE SIMULATION AND ORDERING PROCESS
C
      CALL CHNGE (IEX,NROW,NCOL,COEFF,LCOL,ITAG,LNXT,CE,RE,DE)
      IF (IEX.EQ.1) GO TO 110
      GO TO 120
C
C          WRONG DATA
C
100   WRITE (6,490)
      GO TO 90
C
C          NEW NONZERO ELEMENT IN COEFFICIENT MATRIX
C
110   WRITE (6,500)
      WRITE (6,510)
      READ*,LCAN
      CALL FINDSYM (TEXT,1HC,LCAN)
      IF (LCAN.EQ.1) GO TO 80
      RETURN
120   CONTINUE
C
C          REPEAT BIFACTORIZATION
C
130   IAR=0
C
C          STORE THE NEW VALUES OF THE ELEMENTS OF THE COEFFICIENT
C          MATRICES
C
      DO 140 I=1,LES
      CCE(I)=CE(I)
      RRE(I)=RE(I)
140   CONTINUE
      DO 150 I=1,N
      DDE(I)=DE(I)
150   CONTINUE
      NINT=N
      CALL REDUCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,NINT)
      IF (IAR.EQ.1) RETURN
160   IF (NOQNS.EQ.1) GO TO 230
C
```

```
C         USER SUBROUTINE
C
C         THE USER HAS ACCESS TO THE CURRENT FORM OF THE ANALYZED SYSTEM
C
C         MATRIX V CONTAINS THE CURRENT RIGHT HAND SIDE VECTOR
C
C         CE,DE,RE CONTAIN THE CURRENT COEFFICIENT MATRIX
C
170    IF (ISUB2.GE.1) CALL USUB2 (N,LCOL,ITAG,LNXT,CE,RE,DE,V)
       IF (ISUB2.GT.0) IEND=1
       ISUB2=ISUB2-1
       WRITE (6,520)
       CALL READL (TEXT)
       CALL FINDSYM (TEXT,1HN,LCHE)
       CALL FINDSYM (TEXT,1H*,NOQNS)
       IADR=1
       IF (NOQNS.EQ.1) IADR=2
       IF (LCHE.EQ.1) GO TO 230
       DO 180 J=1,N
       V(J)=VV(J)
180    CONTINUE
       WRITE (6,530)
190    CONTINUE
       READ*,NCHE
       IF (NCHE.GT.0) GO TO 200
       WRITE (6,540)
       GO TO 190
200    IEND=1
       LE=LE+1
       DO 220 J=1,NCHE
210    WRITE (6,550)
       READ*,NROW,COEFF
       IF (NROW.GT.N.OR.NROW.LE.0) GO TO 290
       V(NROW)=COEFF
220    CONTINUE
C
C         SOLUTION WITH NEW RIGHT HAND SIDE VECTOR
C
230    IF (IEND.EQ.0) RETURN
       DO 240 J=1,N
       VV(J)=V(J)
240    CONTINUE
C
C         ADJOINT GRAPH SOLUTION
C
       IF (IAF.EQ.1) GO TO 270
       IF (NOQNS.EQ.1) GO TO 250
       WRITE (6,560)
       CALL READL (TEXT)
C
C         LSOLA REMEMBERS THE CURRENT ORDERING CONCERNING THE SOLUTION
C         OF THE ADJOINT SYSTEM OF EQUATIONS
C
       CALL FINDSYM (TEXT,1HN,LSOLA)
250    IF (LSOLA.EQ.1) GO TO 270
       CALL ASLECN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
       WRITE (6,590) LM,LE
       WRITE (6,570)
       WRITE(6,3^0)
300    FORMAT(1H ,/," SOLUTION",/)
       CALL PRINT (N,V)
       DO 260 J=1,N
       V(J)=VV(J)
260    CONTINUE
270    IF (IAF.EQ.5.AND.IADR.EQ.1) GO TO 10
```

```
      IF (IAF.EQ.5) GO TO 170
      IF (NOQNS.EQ.1) GO TO 280
      WRITE (6,580)
      CALL READL (TEXT)
C
C         LSOL REMEMBERS THE CURRENT ORDERING CONCERNING THE SOLUTION
C         OF THE ORIGINAL SYSTEM OF EQUATIONS
C
      CALL FINDSYM (TEXT,1HN,LSOL)
280   IF (LSOL.EQ.1.AND.IADR.EQ.1) GO TO 10
      IF (LSOL.EQ.1) GO TO 170
      CALL SOLVCN (N,LCOL,NSEQ,ITAG,LNXT,CE,RE,DE,V)
      WRITE (6,590) LM,LE
      WRITE(6,300)
      CALL PRINT (N,V)
      IF (IADR.EQ.1) GO TO 10
      GO TO 170
C
C         WRONG DATA
C
290   WRITE (6,600)
      GO TO 210
C
440   FORMAT (1H ," WILL YOU CHANGE THE COEFFICIENT MATRIX? ",/"  TYPE Y
     1ES OR NO")
450   FORMAT (1H ," NUMBER OF ALTERED COEFFICIENTS")
460   FORMAT (1H ," NUMBER OF CHANGES SHOULD BE A POSITIVE INTEGER")
470   FORMAT (1H ," ENTER ROW AND COLUMN INDEX OF NEW COEFFICIENT")
480   FORMAT (1H ," ENTER NEW COEFFICIENT VALUE")
490   FORMAT (1H ," ELEMENT OUT OF THE MATRIX AREA",/"  WRITE CORRECT VA
     1LUES")
500   FORMAT (1H ," SIMULATION AND ORDERING PROCESS WILL BE COMPLETILY R
     1EPEATED")
510   FORMAT (1H ," TYPE CANCEL IF YOU MADE A MISTAKE")
520   FORMAT (1H ," WILL YOU CHANGE THE RIGHT HAND SIDE VECTOR? ",/"  TY
     1PE YES OR NO")
530   FORMAT (1H ," ENTER THE NUMBER OF CHANGES IN THE RIGHT HAND SIDE V
     1ECTOR")
540   FORMAT (1H ," NUMBER OF CHANGES MUST BE A POSITIVE INTEGER. REENTE
     1R A NEW ONE")
550   FORMAT (1H ," ENTER THE ROW INDEX AND THE VALUE OF THE NEW COEFFIC
     1IENT")
560   FORMAT (1H ," DO YOU NEED AN ADJOINT SOLUTION?",/,"  TYPE YES OR N
     1O")
570   FORMAT (1H ,/," ADJOINT SYSTEM",/)
580   FORMAT (1H ," DO YOU NEED AN ORIGINAL SOLUTION?",/,"  TYPE YES OR
     1NO")
590   FORMAT (1H ," CHANGES IN ELEMENTS",I3," CHANGES IN RIGHT HAND SIDE
     1",I3)
600   FORMAT (1H ," INDEX OUT OF RANGE,",/,"  WRITE CORRECT VALUES")
      END
C
C
      SUBROUTINE FINDSYM (TEXT,SYMB,IFLAG)
C
C         THIS SUBROUTINE FINDS SYMBOL SYMB IN THE TEXT STRING TEXT
C
C         IF SYMBOL HAS BEEN FOUND THE OUTPUT VALUE OF IFLAG
C
C         WILL BE EQUAL 1, OTHERWISE 0
C
      REAL TEXT(1)
      IFLAG=0
      DO 10 I=1,80
      IF (TEXT(I).EQ.1H ) GO TO 10
```

```
      IF (TEXT(I).EQ.SYMB) IFLAG=1 )
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE READL (TEXT)
C
C         THIS SUBROUTINE READS DATA FROM INPUT FILE 4 TO THE
C
C         TEXT MATRIX
C
C         EACH ELEMENT OF THE INPUT FIELD IS STORED IN A SEPARATE
C
C         ELEMENT OF THE TEXT ARRAY
C
      REAL TEXT(1)
      READ (4,10) (TEXT(I),I=1,80) )
      RETURN
C
10    FORMAT (80A1)
      END
C
C
      SUBROUTINE CHNGE (IEX,NROW,NCOL,COEFF,LCOL,ITAG,LNXT,CE,RE,DE)
C
      INTEGER LCOL(1),ITAG(1),LNXT(1)
      COMPLEX COEFF,CE(1),RE(1),DE(1)
C
C         THIS SUBROUTINE CHANGES THE VALUE OF ONE COEFFICIENT IN ROW
C
C         NROW AND COLUMN NCOL OF THE COEFFICIENT MATRIX
C
C         IF THE ELEMENT WAS NOT STORED IN THE COEFFICIENT MATRIX
C
C         OR THE COLUMN OR ROW INDEX IS BEYOND THE MATRIX AREA THEN
C
C         APPROPRIATE INFORMATION IS PRINTED AND NO CHANGE IN COEFFI-
C
C         CIENTS IS MADE
C
C         IN THIS CASE FLAG IEX WILL BE EQUAL 1, OTHERWISE 0
C
      IEX=0
      IF (NROW.EQ.NCOL) DE(NROW)=COEFF
      IF (NROW.EQ.NCOL) RETURN
      IS=LCOL(NCOL)
      IF (IS.EQ.0) GO TO 20
      ISS=IS
10    IF (ITAG(ISS).EQ.NROW) GO TO 50
      IAS=LNXT(ISS)
      IF (IAS.EQ.0) GO TO 20
      ISS=IAS
      GO TO 10
20    IS=LCOL(NROW)
      IF (IS.EQ.0) GO TO 60
      ISS=IS
30    IF (ITAG(ISS).EQ.NCOL) GO TO 40
      IAS=LNXT(ISS)
      IF (IAS.EQ.0) GO TO 60
      ISS=IAS
      GO TO 30
   40 RE(ISS)=COEFF
      RETURN
   50 CE(ISS)=COEFF
```

```
      RETURN
60    WRITE (6,70) NROW,NCOL
      IEX=1
      RETURN
C
70    FORMAT (1H ,/," ELEMENT FROM ROW:",I5," AND COLUMN:",I5," WAS NOT
     1STORED IN THE PREVIOUS COEFFICIENT MATRIX",/)
      END
C
C
      SUBROUTINE PRINTIN(IN)
      INTEGER IN(1)
C
C         PRINT INPUT DATA - VECTOR IN
C
      WRITE(6,100)
  100 FORMAT(///,1H ,"HIERARHICAL ANALYSIS OF LINEAR DECOMPOSED SYSTEMS
     +(CSDSLE PACKAGE)",///)
      WRITE(6,110)
  110 FORMAT(1H ,"INPUT DATA",/,1H ,"----------",/,)
      WRITE(6,120)IN(4)
  120 FORMAT(1H ,"   IN(4) - NUMBER OF SUBGRAPHS WHERE THE SOLUTION WILL
     +BE RECALCULATED ",2(2H .),I5,/)
      WRITE(6,130)IN(5)
  130 FORMAT(1H ,"   IN(5) - NUMBER OF PROPER BLOCKS ",20(2H .),
     +I5,/)
      WRITE(6,140)IN(7)
  140 FORMAT(1H ,"   IN(7) - INDICATOR FOR PRINTING INTERMEDIATE RESULTS
     +",10(2H .),I5,/)
      WRITE(6,150)IN(9)
  150 FORMAT(1H ,"   IN(9) - THE HIGHEST INDEX OF ALL SUBGRAPHS",15(2H .)
     +,I5,/)
      WRITE(6,160)IN(10)
  160 FORMAT(1H ,"   IN(10) - NUMBER OF MODIFIED SUBGRAPHS ",17(2H .),I
     +5,/)
      WRITE(6,170)IN(11)
  170 FORMAT(1H ,"   IN(11) - LENGTH OF BASIC RECORD ",20(2H .),I5,/)
      WRITE(6,180)IN(12)
  180 FORMAT(1H ,"   IN(12) - LENGTH OF ADDRESSING RECORD",18(2H .),I5,/
     +)
      WRITE(6,190)IN(15)
  190 FORMAT(1H ,"   IN(15) - MAXIMUM NUMBER OF BASIC RECORDS",16(2H .),
     +I5,/)
      WRITE(6,200)IN(16)
  200 FORMAT(1H ,"   IN(16) - MAXIMUM NUMBER OF EQUATIONS FOR THE SUBGRAP
     +H ",9(2H .),I5,/)
      WRITE(6,210)IN(17)
  210 FORMAT(1H ,"   IN(17) - MAXIMUM AREA FOR SPARSE MATRIX CALCULATIONS
     +",10(2H .),I5,/)
      WRITE(6,220)IN(18)
  220 FORMAT(1H ,"   IN(18) - NUMBER OF NONZERO ELEMENTS IN THE MAXIMUM S
     +UBMATRIX",6(2H .),I5,/)
      WRITE(6,230)IN(19)
  230 FORMAT(1H ,"   IN(19) - THE LOWEST INDEX OF THE DECOMPOSITION NODES
     +",10(2H .),I5,/)
      WRITE(6,240)IN(20)
  240 FORMAT(1H ,"   IN(20) - NUMBER OF DATA RECORDS ",20(2H .),I5,
     +/)
      WRITE(6,250)IN(22)
  250 FORMAT(1H ,"   IN(22) - INDICATOR FOR RHS UPDATING FORMULA ",14(2H
     +.),I5,/)
      WRITE(6,260)IN(23)
  260 FORMAT(1H ,"   IN(23) - INDICATOR FOR OPERATIONS ON RANDOM FILES ",
     +11(2H .),I5,/)
      WRITE(6,280)IN(25)
```

```
  280 FORMAT(1H ,"   IN(25) - MAXIMUM LENGTH OF DATA AND RHS RECORD",13(2
     +H .),I5,/)
      WRITE(6,270) IN(26)
  270 FORMAT(1H ,"   IN(26) - NUMBER OF ALL DECOMPOSITION NODES",15(2H .)
     +,I5,/)
      M=26+1
      L=26+IN(9)/2
      WRITE(6,290)(IN(I),I=M,L)
  290 FORMAT(1H ,/," NUMBERS OF EXTERNAL NODES IN THE SUBSTITUTE SUBGRAP
     +HS",//,(20I4))
      M=L+1
      L=L+IN(5)
      WRITE(6,300)(IN(I),I=M,L)
  300 FORMAT(1H ,/," INDICES OF ALL PROPER BLOCKS",//,(20I4))
      IF(IN(22).NE.2) RETURN
      M=L+1
      L=L+IN(4)
      WRITE(6,310)(IN(I),I=M,L)
  310 FORMAT(1H ,/," INDICES OF SUBGRAPHS WHERE SOLUTION WILL BE CALCULA
     +TED",//,(20I4))
      RETURN
      END
C
C
      SUBROUTINE PRINTNC(N,NROW,NCOL,V)
C
C         THIS ROUTINE PRINTS THE NONZERO COEFFICIENTS OF A COMPLEX
C
C         MATRIX V
C
      INTEGER NROW(1),NCOL(1)
      COMPLEX V(1)
      IF(N.LE.0) RETURN
      IF(N.GT.1) GOTO 100
      WRITE(6,110)
      WRITE(6,120)
      GOTO 20
100   CONTINUE
      IF (N.EQ.4.OR.N.EQ.2) GO TO 10
      WRITE (6,70)
      WRITE (6,40)
      GO TO 20
10    WRITE (6,60)
      WRITE (6,50)
20    N3=(N+2)/3
      DO 30 I=1,N3
      LI=I
      LO=I+2*N3
      IF (LO.GT.N) LO=I+N3
      IF(LO.GT.N) LO=I
      WRITE (6,80) (NROW(J),NCOL(J),V(J),J=LI,LO,N3)
30    CONTINUE
      RETURN
C
40    FORMAT (1H ,11X,"      REAL           IMAGINARY",18X,
     1"REAL           IMAGINARY",18X,"REAL            IMAGINARY",/)
50    FORMAT (1H ,11X,"      REAL           IMAGINARY",18X,
     1"REAL           IMAGINARY",/)
60    FORMAT (1H ," ROW  COL  ",11X,"VALUE                 ROW  COL"
     1,11X,"VALUE",/)
70    FORMAT (1H ," ROW  COL  ",11X,"VALUE                 ROW  COL  "
     1,11X,"VALUE            ROW  COL  ",11X,"VALUE",/)
80    FORMAT(1X,I4,I5,2E15.6,2I5,2E15.6,2I5,2E15.6)
120   FORMAT(1H ,16X,"REAL",9X,"IMAGINARY",/)
110   FORMAT(1H ," ROW  COL ",14X,"VALUE",/)
```

```
      END
C
C
      SUBROUTINE PRINTRS (N,V,NON)
C
C         THIS ROUTINE PRINTS THE SUBVECTORS OF RHS
C
      INTEGER NON(1)
      COMPLEX V(1)
      IF(N.LE.0) RETURN
      IF(N.GT.1) GOTO 100
      WRITE(6,110)
      WRITE(6,120)
      GOTO 20
100   CONTINUE
      IF (N.EQ.4.OR.N.EQ.2) GO TO 10
      WRITE (6,70)
      WRITE (6,40)
      GO TO 20
10    WRITE (6,60)
      WRITE (6,50)
20    N3=(N+2)/3
      DO 30 I=1,N3
      LI=I
      LO=I+2*N3
      IF (LO.GT.N) LO=I+N3
      IF(LO.GT.N) LO=I
      WRITE (6,80) (NON(J),V(J),J=LI,LO,N3)
30    CONTINUE
      RETURN
C
40    FORMAT (1H ,"                    REAL           IMAGINARY
     1REAL          IMAGINARY                    REAL           IMAGINARY",/)
50    FORMAT (1H ,"                    REAL           IMAGINARY
     1REAL          IMAGINARY",/)
60    FORMAT (1H ," VARIABLE                VALUE                 VARIABLE
     1       VALUE",/)
70    FORMAT (1H ," VARIABLE                VALUE                 VARIABLE
     1       VALUE              VARIABLE                 VALUE",/)
80    FORMAT (1X,3(I6,2X,2E15.6))
120   FORMAT(1H ,15X,"REAL",9X,"IMAGINARY",/)
110   FORMAT(1H ," VARIABLE",13X,"VALUE",/)
      END
C
C
      SUBROUTINE PRINT (N,V)
C
C         THIS ROUTINE PRINTS THE SOLUTION VECTOR
C
      COMPLEX V(1)
      IF(N.LT.1) RETURN
      IF(N.GT.1) GOTO 100
      WRITE(6,110)
      WRITE(6,120)
      GOTO 20
100   CONTINUE
      IF (N.EQ.4.OR.N.EQ.2) GO TO 10
      WRITE (6,70)
      WRITE (6,40)
      GO TO 20
10    WRITE (6,60)
      WRITE (6,50)
20    N3=(N+2)/3
      DO 30 I=1,N3
      LI=I
```

```
      LO=I+2*N3
      IF (LO.GT.N) LO=I+N3
      IF(LO.GT.N) LO=I
      WRITE (6,80) (J,V(J),J=LI,LO,N3)
30    CONTINUE
      WRITE(6,130)
130   FORMAT(///)
      RETURN
C
40    FORMAT (1H ,"                    REAL           IMAGINARY
     1REAL           IMAGINARY                    REAL            IMAGINARY",/)
50    FORMAT (1H ,"                    REAL           IMAGINARY
     1REAL           IMAGINARY",/)
60    FORMAT (1H ," VARIABLE                VALUE                  VARIABLE
     1          VALUE",/)
70    FORMAT (1H ," VARIABLE                VALUE                  VARIABLE
     1          VALUE              VARIABLE                VALUE",/)
80    FORMAT (1X,3(I6,2X,2E15.6))
120   FORMAT(1H ,15X,"REAL",9X,"IMAGINARY",/)
110   FORMAT(1H ," VARIABLE",13X,"VALUE",/)
      END
C
C
      SUBROUTINE PRINTI (N,V,LCOL,ITAG)
C
C         THIS ROUTINE PRINTS COLUMNWISE STORED MATRIX V
C
C         WHICH COLUMNS ARE STORED IN THE AREAS HAVING
C
C         STARTING ADDRESSES LOCATED IN ELEMENTS OF LCOL
C
      COMPLEX V(1)
      INTEGER LCOL(1),ITAG(1)
      IF(N.LT.1) RETURN
      IF(N.GT.1) GOTO 110
      WRITE(6,120)
      WRITE(6,130)
      GOTO 20
110   CONTINUE
      IF (N.EQ.2) GO TO 10
      WRITE (6,80)
      WRITE (6,50)
      GO TO 20
10    WRITE (6,70)
      WRITE (6,60)
20    CONTINUE
      DO 40 II=1,N
      NN=LCOL(II+1)-LCOL(II)
      WRITE (6,100) II
      N3=(NN+2)/3
      DO 30 I=1,N3
      LI=LCOL(II)+I-1
      LO=LI+2*N3
      IF (LO.GE.LCOL(II+1)) LO=LI+N3
      IF(LO.GE.LCOL(II+1)) LO=LI
      WRITE (6,90) (ITAG(J),V(J),J=LI,LO,N3)
30    CONTINUE
40    CONTINUE
      RETURN
C
50    FORMAT (1H ,16X,"REAL",9X,"IMAGINARY",13X,"REAL",9X,"IMAGINARY",13
     1X,"REAL",9X,"IMAGINARY",/)
60    FORMAT (1H ,16X,"REAL",9X,"IMAGINARY",13X,"REAL",9X,"IMAGINARY",/)
70    FORMAT (1H ,"COL.",2X,"ROW",14X,"VALUE",13X,"ROW",14X,"VALUE",/)
80    FORMAT (1H ,"COL.",2X,"ROW",14X,"VALUE",13X,"ROW",14X,"VALUE",13X,
```

```
      1"ROW",14X,"VALUE",/)
 90    FORMAT (5X,3(I5,2E15.7))
 100   FORMAT (1H ,I4)
 130   FORMAT(1H ,16X,"REAL",9X,"IMAGINARY",/)
 120   FORMAT(1H ,"COL.",2X,"ROW",14X,"VALUE",/)
       END
C
C
       SUBROUTINE PRINTS (N,V,LCOL,ITAG,LNXT)
C
C           THIS ROUTINE PRINTS THE SPARSE MATRIX
C
       COMPLEX V(1),COLX(100)
       INTEGER LCOL(1),ITAG(1),LNXT(1),NROW(100)
       IF(N.LT.1) RETURN
       IF(N.GT.1) GOTO 150
       WRITE(6,160)
       WRITE(6,170)
       GOTO 20
 150   CONTINUE
       IF (N.EQ.2) GO TO 10
       WRITE (6,80)
       WRITE (6,50)
       GO TO 20
 10    WRITE (6,70)
       WRITE (6,60)
 20    CONTINUE
       DO 40 II=1,N
       NN=1
       NX=LCOL(II)
       IF(NX.LE.0) GOTO 40
       COLX(NN)=V(NX)
       NROW(NN)=ITAG(NX)
   120 NX=LNXT(NX)
       IF(NX.LE.0) GOTO 110
       NN=NN+1
       COLX(NN)=V(NX)
       NROW(NN)=ITAG(NX)
       GOTO 120
   110 IF(NN.LE.100) GOTO 130
       WRITE(6,140)
       RETURN
   140 FORMAT(1H ,/," MATRIX CONTAINS A COLUMN HAVING MORE THAN 100 ELEME
      +NTS",/)
   130 CONTINUE
       WRITE (6,100) II
       N3=(NN+2)/3
       DO 30 I=1,N3
       LI=I
       LO=LI+2*N3
       IF(LO.GT.NN) LO=LI+N3
       IF(LO.GT.NN) LO=LI
       WRITE(6,90)(NROW(J),COLX(J),J=LI,LO,N3)
 30    CONTINUE
 40    CONTINUE
       RETURN
C
 50    FORMAT (1H ,16X,"REAL",9X,"IMAGINARY",13X,"REAL",9X,"IMAGINARY",13
      1X,"REAL",9X,"IMAGINARY",/)
 60    FORMAT (1H ,16X,"REAL",9X,"IMAGINARY",13X,"REAL",9X,"IMAGINARY",/)
 70    FORMAT (1H ,2X,"COL",2X,"ROW",13X,"VALUE",14X,"ROW",13X,"VALUE",/)
 80    FORMAT (1H ,2X,"COL",2X,"ROW",13X,"VALUE",14X,"ROW",13X,"VALUE",
      114X,"ROW",13X,"VALUE",/)
 90    FORMAT (5X,3(I5,2E15.7))
 100   FORMAT (1H ,I4)
```

```
170     FORMAT( 1H  ,16X, "REAL",9X, "IMAGINARY",/)                  003901
160     FORMAT( 1H  ,2X, "COL",2X, "ROW",13X, "VALUE",/)             003902
        END                                                          003903
```

SOC-307

CSDSLE - A FORTRAN PACKAGE FOR THE SOLUTION OF SPARSE DECOMPOSED SYSTEMS OF LINEAR EQUATIONS

J.A. Starzyk and J.W. Bandler

January 1983, No. of Pages: 212

Revised:

Key Words: Linear systems, sparse matrix solution, decomposed systems, multiple solutions of linear systems

Abstract: CSDSLE is a package of forty-eight subroutines for solving sparse linear equations for the iterative simulation of very large systems. The equations are assumed to be in decomposed form as required by the main subroutine. Zollenkopf's bi-factorization-type algorithm is used to represent the solution of different submatrices. Different options of the package, when utilized properly, may save a significant amount of computer time and memory compared with the standard sparse-matrix subroutines, including those which utilize simple decomposition or block decomposition. The package and documentation have been developed for the CDC 170/730 system with the NOS 1.4 level 552 operating system and the Fortran Extended (FTN) Version 4.8 compiler. The report includes 17 illustrative examples, including the use of mass storage, the implementation of changes in the coefficient matrix and changes in the right-hand side. Local area changes and their effects on the rest of the system are discussed. A comparison with the Harwell package MA28 not using decomposition is reported.

Description: Contains Fortran listing, user's manual. The listing contains 3903 lines, of which 1640 are comments.

Related Work: SOC-289.

Price: $250.00. Source deck or magnetic tape: $500.00. Availability subject to signed author-purchaser agreement.