COMPUTER SOLUTION FOR SPARSE SYSTEMS OF LINEAR EQUATIONS BY SYMBOLIC GENERATION

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Kun Fun Kuo Shang

August, 1975

ACKNOWLEDGEMENT

The author would like to express her gratitude to her advisor, Dr. M. Meicler, for suggesting the thesis topic, continuing guidance in the course of research, and reviewing of the manuscript. A particular appreciation is extended to Dr. A. Newhouse for his comments, criticisms, and suggestions. Also, special thanks are extended to her husband, Henry, who suffered through the whole course with patience and encouragement.

COMPUTER SOLUTIONS FOR SPARSE SYSTEMS OF LINEAR EQUATIONS BY SYMBOLIC GENERATION

An Abstract
of a Thesis
Presented To
The Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

---

by

Kun Fun Kuo Shang

August 1975

ABSTRACT

This paper presents a method of computer solution for sparse systems of linear equations of arbitrary structure by symbolic generation technique.

A computer program, SOURCE, by symbolic processing, generates another program, RESULT, which represents the reduced algorithm. The symbolic generation method used were based on Crout and Gauss-Seidel methods, only non-zero elements are stored and operated on. Because of the increasing use of large order sparse matrices and the tendency to attempt to solve larger order problems, great attention must be focused on core allocation and execution time, as these are the limiting factors that most often dictate the practicality of solving a given problem.

Computer programs were written in FORTRAN V language to implement these methods.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

A sparse matrix is defined to be a matrix containing a high proportion of elements that are zeros. Sparse matrices of large order are of great interest and application in science and industry; for example, electrical networks, structure engineering, power distribution, reactor diffusion and solution to differential equations.

Computation involving sparse matrices have been of widespread use since the 1950s, becoming increasingly popular with the advent of faster cycle times and larger computer memories. Because of the increasing use of large order sparse matrices and the tendency to attempt to solve larger order problems, great attention must be focused on core storage and execution time. Every effort should be made to optimize both computer memory allocation and execution time, as these are the limiting factors that most often dictate the practicality of solving a given problem.

An efficient implementation of a specified method in solving large sparse system of linear algebraic equations of arbitrary structure is described. A computer program SOURCE, by symbolic processing, generates another program RESULT, which represents the reduced algorithm in the sense that only nonzero elements are stored and operated on. In other words, a FORTRAN program SOURCE logically processes each of the scalar formulas defined in a specified method, such as Crout, Gauss-Seidel methods, whenever all terms vanish on the right side of a given formula, SOURCE records this fact; otherwise, it generates a FORTRAN statement. Such a

statement represents a reduced formula in which only nonzero terms are listed. The set of all FORTRAN statements generated constitutes another program RESULT which is a reduced, executable program. In reducing the method for a matrix of ARBITRARY sparseness structure, we want to process each formula individually. This task, while of a simple nature, is extremely tedious, time-consuming, and error-prone when carried out by hand if order N is large. It is best accomplished by a symbolic processing program such as SOURCE.

Two methods are commonly used for the solution of a sparse NxN system of linear equations. One is direct method "CROUT" algorithm, another is iterative method "GAUSS-SEIDEL" algorithm. The efficient implementation of the symbolic processing program that we propose is based on these two algorithms. The reduced algorithms have some important features:

1. Small storage requirements: This small storage requirement results from storing the matrix structure bitwise. If the order of the matrix A is reasonably small, it would make little difference if the full matrix were kept in core. However, if the sparse matrix is of larger order, it becomes efficient in terms of core allocation to store only the nonzero entries of the matrix.

2. Fast processing speed: We can easily see that the arithmetic operations involve in performing a decomposition of a direct method, or one complete cycle of the iterative method, which is at least $\frac{1}{3} N^3$ multiplications. For a large sparse matrix,

of course, the number of multiplications is extremely large. If only the necessary arithmetic operations are carried out, the execution time can be greatly reduced. In our algorithm we only carry out the arithmetic operations on nonzero elements.

3. Uniform linear structure: The program RESULT which SOURCE produces is a long, linear sequence of FORTRAN statements which are simple and all of the same type, therefore execution of the compiled program RESULT is extremely fast.

4. Efficiency to a repetitive problem: In many instances, we must solve system $Ax = b$, a number of times with a fixed sparseness structure but with varying numerical values of the elements of A or b. In this case, our algorithm is particularly efficient when applied to such a repetitive problem, because the symbolic processing by SOURCE is done once and thus the overall efficiency increases as the number of systems to be solved increases.

CHAPTER II

COMPUTER SOLUTION OF REDUCED CROUT ALGORITHM

THE CROUT ALGORITHM AND ITS REDUCTION

We propose a highly efficient algorithm for direct solution of a sparse $N \times N$ system of linear equations

$$Ax = b \tag{2.1}$$

with an arbitrary zero/nonzero structure of A. This algorithm systematically exploits sparseness by eliminating unnecessary arithmetic operations and by storing information in a compact, directly accessible manner. It is based on the CROUT elimination method, which consists of a factoring A into the product, $A = LU$, of a lower triangular matrix L and a unit upper triangular matrix U; and solving by backsubstitution the systems

$$Ly = b \tag{2.2}$$

$$Ux = y \tag{2.3}$$

for y and x, respectively.

To obtain the factors L and U of A , we let m take the value 1,2,..., N successively. For any given m , we compute

$$\ell_{im} = a_{im} - \sum_{h=1}^{m-1} \ell_{ih} u_{hm} , \tag{2.4}$$

for $i = m, m+1, \ldots, N$. Then we successively compute

$$u_{mj} = (a_{mj} - \sum_{h=1}^{m-1} \ell_{mh} u_{hj}) / \ell_{mm} , \tag{2.5}$$

for $j = m+1, m+2, \ldots, N$. In the backsubstitution, we compute

$$y_i = (b_i - \sum_{h=1}^{i-1} \ell_{ih} y_h)/\ell_{ii} , \qquad (2.6)$$

successively for $i = 1,2,\ldots,N$, and

$$x_i = y_i - \sum_{h=i+1}^{N} u_{ih} x_h , \qquad (2.7)$$

for $i = N, N-1,\ldots,1$

Considering a decomposition process, in order to simplify formula (2.4) and (2.5), the first step is to determine which elements of L and U are nonzero. The reduction consists of removing all terms containing zero operands. The equations (2.4) and (2.5) are processed in pairs of batches; for each m the two batches consist of all formulas (2.4) and (2.5), respectively. Processing of the u-batch is analogous to processing of the ℓ-batch and thus we only describe the latter.

The ℓ-batch for $m = 1$ is simple to process since no product terms are involved. For a specific i, a formula is written if and only if a_{i1} is nonzero. For $m \geq 2$, we produce a formula for ℓ_{im} if and only if a_{im}, or at least one of the products in (2.4) is nonzero. Product terms are eliminated in two steps. First, we preserve only those products in which $u_{hm} \neq 0$; this selection is independent of i and thus is made only once per ℓ-batch. Second, for each i we inspect the products selected in the first step and retain only those for which $\ell_{ih} \neq 0$.

The reduction of (2.6) and (2.7) is made by backsubstitution of the system.

IMPLEMENTATION OF THE EFFICIENT METHOD

The efficient generation of the reduced CROUT algorithm by the symbolic program SOURCE is essential for making this algorithm practical. In fact, unless the program can be created and compiled in a reasonable amount of time, the cost of obtaining it may offset the gain we achieve by exploiting sparseness. There are several efforts implemented to overcome it. The symbolic method was originated at the IBM Research Center (see 3, p. 87).

The important aspect is the efficient use of storage, since the processing time is greatly reduced if the entire program can reside in core. In this respect, it is useful to associate with A its adjacency matrix $\hat{A} = (\hat{a}_{ij})$ which is defined by

$$\hat{a}_{ij} = \begin{cases} 1 \text{ if } a_{ij} \neq 0 , \\ 0 \text{ if } a_{ij} = 0 . \end{cases}$$

Since $\hat{A}$ completely characterizes the sparseness structure of A, we need only $\hat{A}$ to generate the reduced CROUT algorithm and $\hat{A}$ can be stored bitwise. Furthermore, we consider the triangular matrices L and U as making up one square matrix C whose elements are

$$\hat{c}_{ij} = \begin{cases} 1_{ij} \text{ if } i \geq j , \\ u_{ij} \text{ if } i < j . \end{cases}$$

Again, as far as symbolic processing is concerned, we only deal with $\hat{C}$, the adjacency matrix of C.

One order of storing the bits of $\hat{A}$ is characterized by the mapping $K(i,j)$ of all pairs (i,j), $1 \leq i, j \leq N$, onto the set of integers $1,2,3,\ldots,N^2$ and is represented in the following diagram:

$$\begin{bmatrix} 1 & N+1 & N+2 & \ldots & 2N-1 \\ 2 & 2N & & \ldots & 4N-4 \\ \vdots & & & & \\ N & 3N-2 & & \ldots & N^2 \end{bmatrix}$$

We start counting down column 1, then across the rest of row 1, down the rest of column 2, across the rest of row 2, etc., until the full NxN array is exhausted. It can be defined as follows:

$$K(i,j) = \begin{cases} (j-1)\ (2N-j) + i \ \text{if} \ i \geq j\ , \\ i(2N-i) - N + j \ \text{if} \ i < j\ . \end{cases} \qquad (2.8)$$

The matrix $\hat{C}$ is stored in the same order as $\hat{A}$. Once $\hat{c}_{ij}$ has been determined, $\hat{a}_{ij}$ is no longer needed and $\hat{c}_{ij}$ is stored in its place. Therefore, the total number of b-bit words required to store $\hat{A}$ and $\hat{C}$ is $(N^2/b)+1$ words (b is a 36-bit word).

The symbolic program SOURCE scans the formulas (2.4) and (2.5) in the CROUT algorithm according to its natural order. It produces a reduced formula which is written as a single FORTRAN statement in terms of just two symbols, A and C, which are one-dimensional arrays of length n and n'. These arrays contain just the n and n' nonzero elements of matrix

A and C, respectively, in a "collapsed" one-dimensional form. The nonzero elements are arranged into an array according to its natural order, it simplifies the compilation of RESULT and economizes on storage. A convenient choice for the collapsed label of a nonzero element a_{ij} is I, defined as the number of nonzero elements encountered in processing the first $K(i,j)$ factorization formulas. Similarly, a natural choice for collapsed label of a nonzero element c_{ij} is J, the number of nonvanishing L-U formulas which have been generated during processing of the first $K(i,j)$ of those formulas. The collapsed notation for a nonzero element a_{ij} and c_{ij} is then $A(I)$ or $C(J)$, respectively. It can be defined as two integer-valued functions of K, $K=1,1,\ldots,N^2$ by

$$I(K) = \sum_{h=1}^{K} \hat{a}_h \,, \qquad J(K) = \sum_{h=1}^{K} \hat{c}_h \,. \tag{2.9}$$

The use of collapsed labels complicates SOURCE, when SOURCE scans through formulas (2.4) for a specific i, it involves those products $\ell_{ih}u_{hm}$, $h=1,2,\ldots,m-1$, in which $u_{hm} \neq 0$. From these products we can easily select those for which; $\ell_{ih} \neq 0$ by refering to the adjacency matrix $\hat{C}$. The problem is then to determine the label of ℓ_{im} and a_{im}, and to retrieve the labels of ℓ_{ih} and u_{hm}. These labels were determined during the processing of the h^{th} batches and are no longer directly accessible. In principle, we could build a correspondence table $j \leftrightarrow K(i,j)$. However, such a procedure would spend an excessive amount of time searching for each product term in every formula. The storage required for the reference table would be excessive which may offset all gains from this algorithm.

The program SOURCE uses nine sequences of integers of length at most N in generating the reduced algorithm. Some of the sequences are used for obtaining the collapsed formula labels, other sequences give the key information which is needed to generate the reduced formulas by a simple indexing technique.

DESCRIPTION OF THE SYMBOLIC PROCESSING

The generation of the reduced CROUT algorithm by SOURCE is achieved recursively at three different levels, while the program SOURCE scans the formulas (2.1) and (2.2) in the CROUT algorithm according to its natural order. The top level is with respect to a batch number m, the intermediate level is with respect to the elements within a given batch, and the lowest level is with respect to the summation over the product terms in processing the formula associated with a given element.

The program SOURCE performs its task by simple indexing technique, making us of 14 sequences of integers, four of these are needed throughout the processing of the entire matrix. The other ten are kept and updated during processing of a pair of batches, five of these are used in generating reduced ℓ-formulas, the other five in generating u-formulas. Since the ℓ-batch and u-batch are processed consecutively only five of the ten sequences are needed at a time, thus storage is required for only nine sequences. We denote the sequences by the mnemonic symbols $c, cc, r, rc, bcc, b, bcr, ecc, ecr$, the letter c in the first position refer to a list of addresses of u-elements in a column and it is used to process the ℓ-elements of the same column. Similarly, r in the first position

indicates a list of addresses of ℓ-elements in a row used to process the u-elements of that same row. In addition, sequences whose symbols begin with r or c are needed during the processing of the entire matrix. The symbols associated with the processing of an individual batch or element begin with the letters b or e, respectively. The presence or absence of the letter c in the second position indicates a collapsed or noncollapsed label, respectively. Finally, the letter c or r in the third position denotes levels of column or row elements, respectively.

Before further description of the algorithm, it is important to have the following sequences defined.

$$c(v) = K(v,m) \qquad v = 1,2,\ldots,m-1$$

where c refers to a list of addresses of u-elements in a column with respect to the batch m. It is used to process the ℓ-elements of the same column.

$$cc(v) = J(c(v)-1) \qquad v = 1,2,\ldots,m-1$$

where cc refers to a list of the collapsed labels of u-elements in a column with respect to the batch m.

$$r(v) = K(m,v) \qquad v = 1,2,\ldots,m-1$$

where r refers to a list of addresses of ℓ-elements in a row with respect to batch m. This list is used to process the u-elements of the same row.

$$rc(v) = J(r(v)-1) \qquad v = 1,2,\ldots,m-1$$

where rc refers to a list of the collapsed labels of ℓ-elements in a row with respect to the batch m. And $K(i,j)$, $I(K)$, $J(K)$ are defined by (2.8) and (2.9).

For simplicity, the dependence of these symbols on m is not explicitly indicated. Figures 2.1 and 2.2 depict one-dimensional, as well as the natural two-dimensional, ordering of the compacted L-U matrix, respectively.

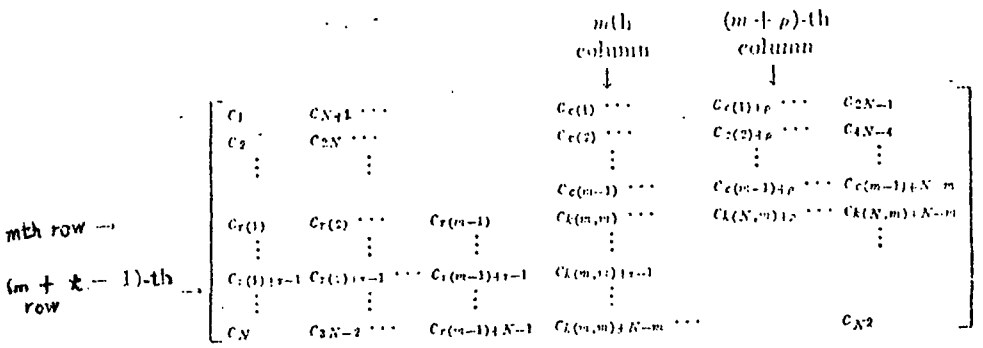


Figure 2-1

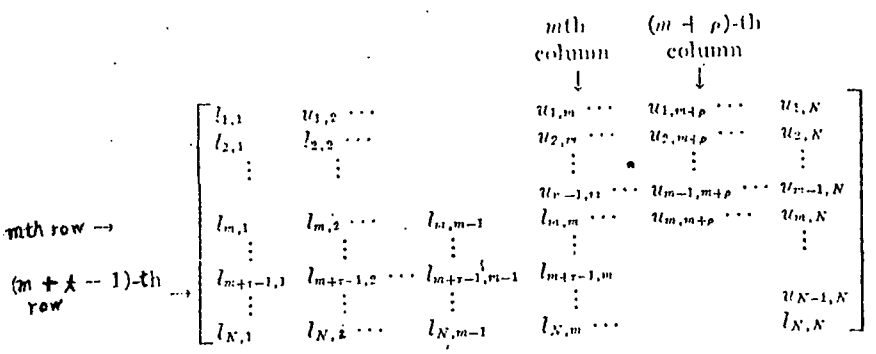


Figure 2-2

First, the symbolic program SOURCE scans the formulas (2.1), (2.2), and (2.3) for the batch m under collapsed one-dimensional form

$$C_K(m,m)+t-1 = - \sum_{v=1}^{m-1} C_{r(v)+t-1}\, C_{c(v)} + a_{K(m,m)+t-1} \qquad (2.10)$$

for $t = 1,2,\ldots,N-m+1$

$$Y_m = (-\sum_{v=1}^{m-1} C_{r(v)}\, Y_v + b_m) \,/\, C_{K(m,m)} \tag{2.11}$$

$$C_K(N,m)+p = (-\sum_{v=1}^{m-1} C_{r(v)}\, C_{c(v)+p} + a_{K(N,m)+p}) \,/\, C_{K(m,m)} \tag{2.12}$$

for $p = 1,2,\ldots,N-m$

Since the matrix A is sparse, there are some zero terms on the right side of the above formulas. We only need the nonzero terms in order to eliminate the unnecessary arithmetic operations. We consider the monotonically increasing sequence $\{v(g),\ g=1,\ldots,g_{max}\}$ (which may be empty), which consists of row numbers of 1's among the $C_{K(v,m)}$, $v=1,2,\ldots,m-1$ (i.e., of the nonzero u-elements in the m^{th} column above the main diagonal). Let

$$bcc(g) = J(c(v(g))) \tag{2.13}$$

denote the collapsed one-dimensional label of the nonzero element as $u_{v(g),m}$. Also, let

$$b(g) = r(v(g)) = K(m,v(g)) \tag{2.14}$$

denote the K-label (uncollapsed one-dimensional label) of the location as $(m,v(g))$. Finally, set

$$bcr(g) = J(r(v(g))-1) \tag{2.15}$$

The mapping (2.8) is so constructed that for $t=1,2,\ldots,N-m+1$, we have $K(m+t-1,v(g)) = K(m,v(g)) + t-1 = r(v(g)) + t-1 = b(g) + t-1$, thus, the sum appearing in (2.1) which can be written as $\sum_{v=1}^{m-1} \ell_{m+t-1}\, u_{v,m}$ and is equivalent to the sum in (2.10), can be collapsed to

$$\sum_{g=1}^{g_{max}} \ell_{m+t-1,v(g)}\, u_{v(g),m} = \sum_{g=1}^{g_{max}} C_{b(g)+t-1}\, C_{c(v(g))} \qquad (2.16)$$

Thus, knowing $b(g)$ allows us to determine directly (via C) the zero/non-zero structure of the subset of elements $\ell_{m+t-1,v(g)}$, $g=1,\ldots,g_{max}$. for $t=1,1,\ldots,N-m+1$. The number of nonvanishing reduced ℓ-u formula generated in processing all formulas up to and including that for $\ell_{m-1,v(g)}$ is $bcr(g)$. The value of $b(g)$ and of $bcr(g)$ are used jointly to obtain directly the collapsed labels of the nonzero elements in the above mentioned subset.

By (2.13), equation (2.10) can be written in semi-collapsed form as

$$C_{K(m,m)+t-1} = a_{K(m,m)+t-1} - \sum_{g=1}^{g_{max}} C_{b(g)+t-1}\, C_{bcc(g)} \qquad (2.17)$$

To finish the processing of formula (2.17) for a given t, we examine $\hat{C}_{b(g)+t-1}$ for $g=1,\ldots,g_{max}$. In case $\hat{C}_{b(g)+t-1} = 1$, we must replace the uncollapsed label $b(g)+t-1$ by the corresponding collapsed label. If $\hat{C}_{b(g)+t-1} = 0$, we delete a term in the sum of the products.

Finally, we determine the collapsed labels of $a_{K(m,m)+t-1}$ and $C_{K(m,m)+t-1}$ (if they exist) and write out the resulting formula (if they exist).

We can write formula (2.17) into a simpler form by further examining $C_{b(g)+t-1}$, $t=1,2,\ldots,N=m+1$. To implement this step, we consider a monotonically increasing sequence $\{g(s),\ s=1,\ldots,s_{max}\}$ (which may be empty), consisting of the column numbers of 1's among $\{\hat{C}_{K(m+p,v(g))},\ g=1,\ldots,g_{max}\}$ (i.e. of those nonzero ℓ-elements in the $(m+p)th$ row whose column indices belong to the set $v(g)$. The values $g(s)$ constitute a subsequence of the subsequence $v(g)$. For $s=1,\ldots,s_{max}$, we set

$$ecr(s) = bcr(g(s)) + 1 = J(r(v(g(s))) + p) \tag{2.18}$$

and

$$ecc(s) = bcc(g(s)) = J(c(v(g(s)))) \tag{2.19}$$

where

$$s_{max} = \sum_{g=1}^{g_{max}} \hat{C}_{b(g)} = \sum_{v=1}^{m-1} \hat{C}_{r(v)+p}\, \hat{C}_{c(v)}$$

By formulas (2.18) and (2.19), equation (2.17) can be written in a collapsed form simply as

$$C_J = \sum_{s=1}^{s_{max}} C_{ecr(s)}\, C_{ecc(s)} + a_I$$

or

$$C_J = - \sum_{s=1}^{s_{max}} C_{ecr(s)}\, C_{ecc(s)} \quad if \quad a_{m+t-1,m} = 0 .$$

or

$$C_J = a_I \quad if \quad s_{max} = 0$$

We have completed the description of the process for $LY = b$. The reduction process for $UX = Y$ is analogous to that of $LY = b$.

REDUCTION ALGORITHM

After reading the input data and initializing the bit map list, the program enters the reduction algorithm, which is written in program notations, as detailed below:

1. Determine the sequences *bcc*, *b*, and *bcr*, simultaneously updating *cc(v)*, *v=1,1,...,m-1* for the *m*th batches.

2. Determine *r(m)*, *rc(m)*, and *jd* (*jd* is the collapsed formula label of the diagonal element $\ell_{m,m}$).

3. Process the ℓ-formula (2.10) and write out a reduced Crout formula in a collapsed form.

4. Determine the sequences *bcr*, *b*, and *bcc*, simultaneously updating *rc(v)*, *v=1,2,...,m-1*. Write out the formula for y_m.

5. Determine *c(m)* and *cc(m)*.

6. Process the u-formula (2.12) and write out a reduced Crout formula in a collapsed form.

7. Update the sequences $c(v)$ and $r(v)$, $v=1,2,\ldots,m-1$.

8. If $m<N$ then $m = m+1$, go to step 1, otherwise stop and a symbolic reduced program RESULT has been generated.

PROGRAM INPUT/OUTPUT AND EXAMPLE

The program SOURCE is written in FORTRAN V and runs on the Univac 1108 series. It will handle matrices up to order $N = 600$. Input to the program is by cards as follows: (1) one card containing N, the order of the given matrix A; and (2) $[NUM/13]+1$ cards (NUM, the number of nonzero elements of A) locating the nonzero elements by the one-dimensional labeling k, $k = 1,2,\ldots,N^2$, defined by equation (2.11).

After reading the input data and storing the sparseness structure of A bitwise, the program enters the reduction algorithm described in Chapter II.

The main part of the output is the symbolicly generated FORTRAN program RESULT, which represents the reduced efficient CROUT method, and is written on the disk.

The input to RESULT consists of two one-dimensional arrays, the first array represents the numerical values of the nonzero elements of A ordered in accordance with the one-dimensional labeling as aforementioned. The second array consists of the elements of b. The output of RESULT is the solutions of equations (2.1).

To illustrate the use of our algorithm, we give a complete description of its application to the following small system of equations:

$$\begin{bmatrix} 7 & 0 & -3 & 0 & -1 & 0 \\ 2 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -3 & 0 & 0 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} -11 \\ 38 \\ 0 \\ 13 \\ 11 \\ -22 \end{bmatrix}$$

In this case, the input to SOURCE is

```
6
1   2   4   8   10   12   15   21   28   30   33   36
```

The program SOURCE which was written in FORTRAN V and is listed in the Appendix, program 1. The generated symbolic program RESULT is shown in Figure 2.3.

The numerical values of the nonzero elements of A, which are input to RESULT, are

```
7.0000000+00  2.0000000+00  -3.0000000+00  -3.0000000+00  -1.0000000+00
8.0000000+00 -1.0000000+00   1.0000000+00   5.0000000+00  -2.0000000+00
```

and the elements of b are

```
-1.1000000+01  3.8000000+00  0.0000000  1.3000000+01  1.1000000+01
-2.2000000+01
```

After the execution of program RESULT, we obtain the solutions of equations (2.1)

```
x(1) = -9.9999961-01
x(2) =  4.9999999+00
```

x(3) = 0.0000000

x(4) = 2.0000002+00

x(5) = 4.0000000+00

x(6) = -2.9999999+00

To explicitly discuss the reduction algorithm, we give a complete description of its process to the above small system of equations.

As an example, we discuss the batch processing for $m = 1,2$.

At the beginning, the adjacency information grouped with respect to the batch number $m = 1$ is:

11010001010

and initialize $K = 0,\ I = 0,\ J = 0$

In step 1, we obtain

$$c(1) = 0,\ cc(1) = 0,\ maxc = 0,\ K = K+1 = 1$$

In step 2, we update sequences $r,\ rc$

$$r(1) = K+1 = 2,\ rc(1) = J+1 = 1,\ jd = J+1 = 1$$

In step 3, since $maxc = 0$, all elements above the diagonal in column 1 are zero. Reduced formulas are generated according to (2.10) and update $I,\ J,\ K$.

$$I = I+1 = 1,\ J = J+1 = 1$$

since $\hat{a}(1) = 1,\ C(J) = A(I),\ i.e.\ C(1) = A(1)$

To continue this processing until $t = N-m+1$, we obtain

since $\hat{a}(2) = 1$, *thus*, $I = I+1 = 2$, $J = J+1 = 2$, $C(2) = A(2)$

since $\hat{a}(4) = 1$, *thus*, $I = I+1 = 3$, $J = J+1 = 3$, $C(3) = A(3)$

$K = K+t-1 = 6$

In step 4, since $m = 1$, the sum in (2.11) vanishes. A simple reduced formula generated according (2.11),

$y(m) = B(m)/C(jd)$, *i.e.* $y(1) = B(1)/C(1)$

$K = K+1 = 7$

In step 4, we update sequences c, cc

$c(1) = K = 7$, $cc(1) = J = 3$

In step 6, since $maxr = 0$, all elements to the left of the diagonal in row 1 are zero. Reduced formulas are generated according to (2.12).

$I = I+1 = 4$, $J = J+1 = 4$

since $\hat{a(8)} = 1$, $C(J) = A(I)/C(jd)$ *i.e.* $C(4) = A(4)/C(1)$

To continue this processing until $p = N-m$.

$I = I+1 = 5$, $J = J+1 = 5$

since $\hat{a(10)} = 1$, $C(J) = A(I)/C(jd)$, *i.e.* $C(5) = A(5)/C(1)$

$K = K+p-1 = 11$

In step 7, since $m = 1$, no processing is needed. After the complete processing for batch $m = 1$, the program enters batch number $m = 2$. The adjacency information will be updated and appears as

11010001010100100Ø0Ø0+

+The symbol Ø represents fill-in; i.e. an originally zero element became non-zero during processing.

In step 1, since $\hat{c}(7) = 0$, we have $maxc = 0$, and no intermediate sequences are generated.

$$K = K+1 = 12$$

In step 2, we update r, rc

$$r(2) = K+1 = 13,\ rc(2) = J+1 = 6,\ jd = J+1 = 6$$

In step 3, since $maxc = 0$, all elements above the diagonal in column 2 are zero. Reduced formulas are generated according to (2.10) and update I, J, K

$$I = I+1 = 6,\ J = J+1 = 6$$

since $\hat{a}(12) = 1,\ C(J) = A(I),\ \ i.e.\ C(6) = A(6)$

To continue this processing until $t = N-m+1$.

$$I = I+1 = 7,\ J = J+1 = 7$$

since $\hat{a}(15) = 1,\ C(J) = A(I),\ \ i.e.\ C(7) = A(7)$

$$K = K+t-1 = 16$$

In step 4, since $\hat{c}(2) = 1$, we update sequence rc

$$rc(1) = rc(1) + 1 = 2,\ K = K+1 = 17$$

and form the intermediate level labels

$$bcr(1) = rc(1) = 2,\ bcc(1) = cc(1) = 3$$

and form the bottom level labels

$$ecr(1) = rc(1) = 2,\ ecc(1) = v = 1$$

then write the backsubstitute formula

$$y(m) = (-C(ecr(1)*y(ecc(1) + B(m)) / C(jd)\ \ i.e.$$

$$y(2) = (-C(2)*y(1) + B(2)) / C(6)$$

In step 5, we determine the top-level labels

$$c(2) = K = 17,\ cc(2) = J = 7$$

In step 6, since $\hat{c}(18) = 1$, we form

$$bcc(1) = bcc(1) + 1 = 4$$

$$ecr(1) = bcr(1) = 2$$

$$ecc(1) = bcc(1) = 4$$

We update J by $J+1$, $J = 8$, and write the reduced formula according to (2.12).

$$C(J) = (-C(ecr(1)*C(ecc(1)) /C(jd) \qquad i.e.$$

$$C(8) = (-C(2)*C(4)) / C(6)$$

Next, since $\hat{c}(10) = 1$, we form

$$bcc(1) = bcc(1)+1 = 5$$

$$ecr(1) = bcr(1) = 2$$

$$ecc(1) = bcc(1)\ 5$$

$$J = J+1 = 9$$

$$C(J) = (-c(ecr(1)*C(ecc(1)) / C(jd) \qquad i.e.$$

$$C(9) = (-C(2)*C(5)) / C(6)$$

In step 7, we update c, r

$$c(1) = c(1)+1 = 8$$

$$r(1) = r(1)+1 = 3$$

A complete list of FORTRAN statements is shown in Figure 2.3.

```
      DIMENSION A(    12),B(     6),C(    18),Y(     6),X(     6)
    5 READ(5,10,END=99) (A(I),I=1,    12)
   10 FORMAT(12F5.4)
      WRITE (6,20) (A(I),I=1,    12)
   20 FORMAT(1X12F10.4)
      READ (5,10) (B(I),I=1,     6)
      WRITE(6,20) (B(I),I=1,     6)
      C(     1)=A(     1)
      C(     2)=A(     2)
      C(     3)=A(     3)
      Y(     1)=B(     1)/C(     1)
      C(     4)=A(     4)/C(     1)
      C(     5)=A(     5)/C(     1)
      C(     6)=A(     6)
      C(     7)=A(     7)
      Y(     2)=(-C(     2)*Y(     1)
     *+B(     2))/C(     6)
      C(     8)=(-C(     2)*C(     4)
     *)/C(     6)
      C(     9)=(-C(     2)*C(     5)
     *)/C(     6)
      C(    10)=A(     8)
      C(    11)=-C(     3)*C(     4)
      C(    12)=-C(     7)*C(     8)
      Y(     3)=B(     3)/C(    10)
      C(    13)=A(     9)
      C(    14)=A(    10)
      Y(     4)=(-C(     3)*Y(     1)-C(    11)*Y(     3)
     *+B(     4))/C(    13)
      C(    15)=(-C(     3)*C(     5)
     *)/C(    13)
      C(    16)=A(    11)-C(     7)*C(     9)
      C(    17)=-C(    14)*C(    15)
      Y(     5)=(-C(     7)*Y(     2)-C(    12)*Y(     3)
     *+B(     5))/C(    16)
      C(    18)=A(    12)
      Y(     6)=(-C(    14)*Y(     4)-C(    17)*Y(     5)
     *+B(     6))/C(    18)
      X(     6)=Y(     6)
      X(     5)=Y(     5)
      X(     4)=Y(     4)-C(    15)*X(     5)
      X(     3)=Y(     3)
      X(     2)=Y(     2)-C(     9)*X(     5)
     *-C(     8)*X(     3)
      X(     1)=Y(     1)-C(     5)*X(     5)
     *-C(     4)*X(     3)
      WRITE(6,20) (C(I),I=1,    18)
      WRITE(6,20) (Y(I),I=1,     6)
      WRITE(6,20) (X(I),I=1,     6)
      GO TO 5
   99 END
```

Figure 2.3

DISCUSSION

In performing the decomposition $LU = A$, we have ignored the possibility of a breakdown of the process due to one of the diagonal elements, usually of L, having the value zero, so that the subsequent division by this value is prohibited.

For example the matrix

$$\begin{bmatrix} 4 & 3 & 2 & 2 \\ 1 & -1 & 4 & 4 \\ 2 & -2 & 8 & -6 \\ 1 & -1 & -3 & 4 \end{bmatrix}$$

has the perfectly good inverse

$$\frac{1}{14}\begin{bmatrix} 2 & -2 & 2 & 4 \\ 2 & 0 & -2 & -4 \\ 0 & 2 & 0 & -2 \\ 0 & 2 & -1 & 0 \end{bmatrix}$$

but the matrix of order three obtained by omitting the last row and column of A is singular, and we find

$$\underset{B}{\begin{bmatrix} 4 & 3 & 2 \\ 1 & -1 & 4 \\ 2 & -2 & 8 \end{bmatrix}} = \underset{L}{\begin{bmatrix} 4 & & \\ 1 & -1.75 & \\ 2 & -3.5 & 0 \end{bmatrix}} \underset{U}{\begin{bmatrix} 1 & 0.75 & 0.5 \\ & 1 & -2 \\ & & 1 \end{bmatrix}}$$

We have already seen that if the last diagonal element of L vanishes, the determinant of the matrix A is zero, the matrix is singular, and there is no unique solution of the linear equations involving matrix A. If an earlier diagonal term vanishes, however,

it means only that a principal sub-matrix A has a zero determinant, and does not necessarily imply singularity of A. Usually these difficulties can be avoided by using partial positioning for size. We achieve this by the simple device of keeping track at each stage of which row has the largest element in the relevant column. But in the symbolic reduced algorithm as far as symbolic processing is concerned, we only deal with $\hat{A}$, the adjacency matrix of A. The numerical values of A are invisible to the program SOURCE, and the reduced method will fail, if any sub-matrix of A has a zero determinant, even if the matrix A is nonsingular. The program SOURCE proceeds under the assumption only that the diagonal element of C are all nonzero.

The entire program SOURCE requires only N^2 bits for the bitmap, $9N$ words for keeping track of key information and 5000 words of program itself. This compares to the total need of core allocation of the Crout algorithm which is at least N^2 words. Thus, it makes quite a difference if the order of the matrix is larger than 150. Figure 2.6 shows the comparison of core allocation.

Another major consideration is the difference in execution time. A set of examples have been tested to solve an unsteady-state heat flow problem with fixed boundary conditions, we have solved a linear system of order $N = 10, 25, 50, 75, 100, 125, 150, 175, 200, 225$ with a sparse matrix of the structure indicated in Figure 2.4. Table 2.1 tabulates the execution time and Figure 2.5 depicts the execution time for the two algorithms.

$$\begin{bmatrix} (1-2r) & r & & & \\ r & (1-2r) & r & & \\ & \cdot & & & \\ & & \cdot & & \\ & & & r & (1-2r) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_i \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \cdot \\ 1 \end{bmatrix}$$

Let $r = 1/4$

Figure 2.4

Table 2.1 Comparison of Execution Time Between Crout Algorithm and Reduced Crout

Algorithm / CPU Time (sec) / Order of Matrices	Crout	Reduced Crout	Result
10	.124	2.636	.062
25	.417	4.109	.152
50	1.036	7.590	.274
75	2.900	11.548	.368
100	8.913	15.273	.490
125	13.184	18.475	.655
150	22.745	23.547	.739
175	36.374	26.793	.923
200	53.814	33.950	.969
225	74.036	41.344	1.073

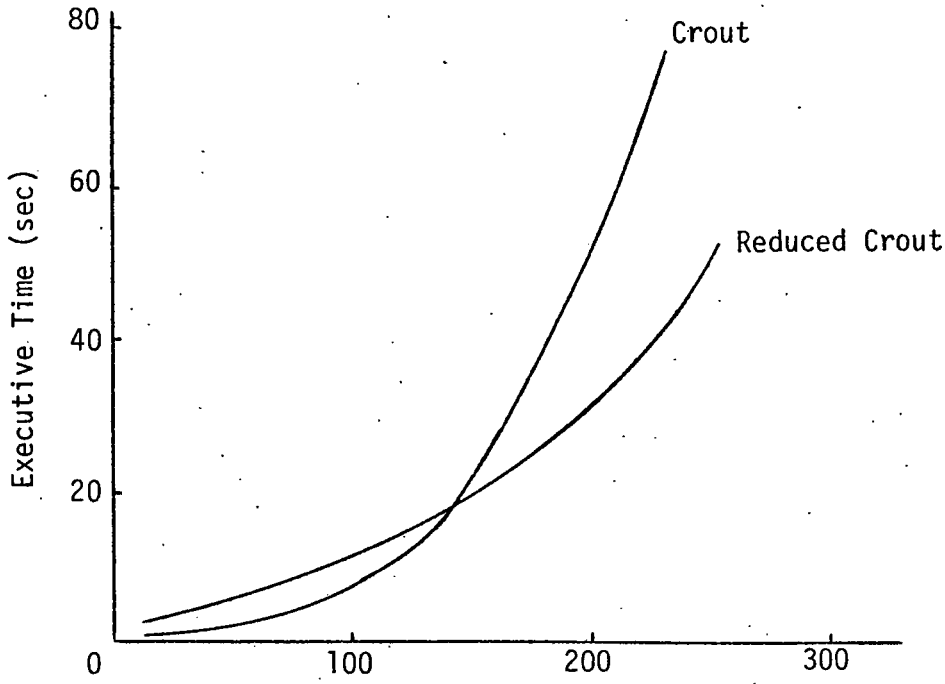


Figure 2.5

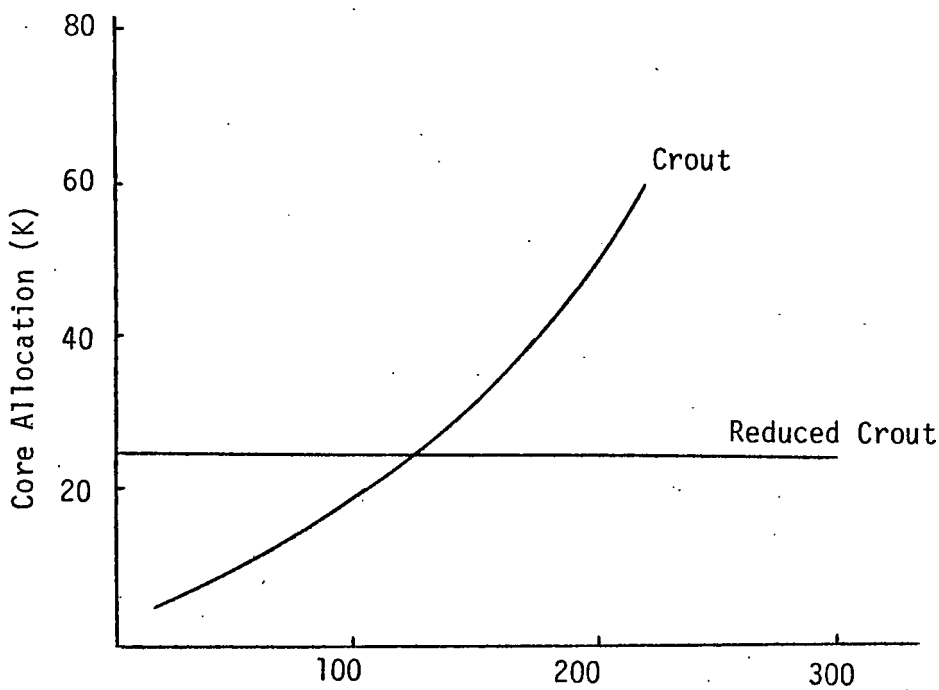


Figure 2.6

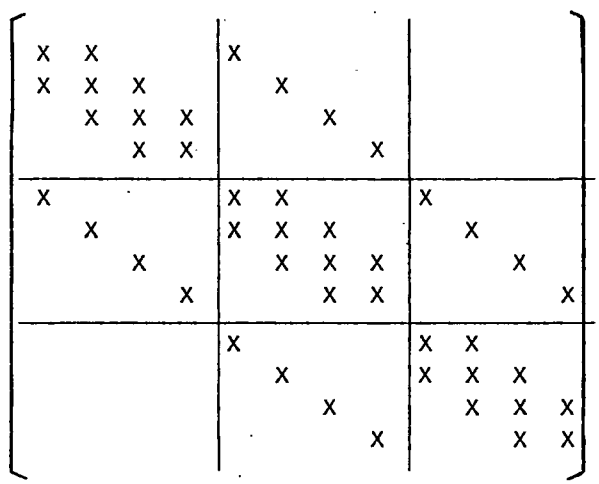
CHAPTER III

COMPUTER SOLUTION BY REDUCED GAUSS-SEIDEL ALGORITHM

THE GAUSS-SEIDEL ALGORITHM AND ITS REDUCTION

We propose another efficient algorithm for the iterative solution of a sparse *NxN* system of linear equations. Iterative methods are used mainly in those problems for which convergence is known to be rapid, so that the solution is obtained with less work than that of a direct method, and for matrices of large order but with many zero elements, the so-called "sparse" matrices, for which elimination methods would be very laborious and need much storage.

An important example comes from finite difference methods for solving elliptic differential equations. The resulting matrix may have a form like



and be of very large order. It is clear that any elimination method would rapidly 'fill up' the places where elements are initially zero in the region enclosed within the sloping lines bounding the array.

We now turn to a consideration of an iterative method for solving linear equations. This algorithm systematically exploits sparseness by eliminating unnecessary arithmetic operations and by storing information in a compact, directly accessible manner. This is based on the

Gauss-Seidel method, we write the matrix A in the form, $A = D + L + U$, where D is a diagonal matrix and L and U are, respectively, lower and upper triangular matrices with zeros on the diagonal. Then (2.1) may be written

$$X_{i+1} = -D^{-1}(LX_{i+1} + UX_i) + D^{-1}b$$

Denote the j^{th} component of X_i by X_i^j. To compute X_{i+1}^1 we use the first equation in the form

$$X_{i+1}^1 = -(1/a_{11})\left(\sum_{j=2}^{N} a_{1j}X_i^j - b_1\right)$$

In general

$$X_{i+1}^r = -(1/a_{rr})\left(\sum_{j=1}^{r-1} a_{rj}X_{i+1}^j + \sum_{j=r+1}^{N} a_{rj}X_i^j - b_r\right) \qquad (3.1)$$

for $r = 1,2,\ldots,N$

The first step is to determine which elements of L and U are nonzero. The reduction consists of removing all terms containing zero operands. The equation (3.1) is processed with respect to the batch number r, product terms are eliminated in one step, we preserve only those products in which $a_{rj} \neq 0$; a reduced formula is written after each batch processing of equation (3.1).

IMPLEMENTATION OF THE EFFICIENT METHOD

The efficient generation of the reduced Gauss-Seidel algorithm by the symbolic program SOURCE1 is essential for making this program practical. In fact, it is the same as in the implementation of the reduced Crout algorithm. There are several additional points that need to be clarified.

The important aspect is the efficient use of storage, we store only the nonzero elements. One order of storing the nonzero elements is characterized by the mapping $K(i,j)$ of all pairs (i,j), $1 \leq i, j \leq N$, onto the set of integers $1, 2, 3, \ldots, N^2$ and is represented in the following diagram:

$$\begin{bmatrix} 1 & 2 & \cdots & N \\ N+1 & N+2 & \cdots & 2N \\ 2N+1 & & \cdots & \\ \vdots & & & \\ (N-1)N+1 & & \cdots & N^2 \end{bmatrix}$$

We start counting across row 1, then the second row, etc., until the full NxN array is exhausted. It can be defined as follows:

$$K(i,j) = (i-1) * N + J$$

$j = 1,2,\ldots,N$ for each i which $i = 1,2,\ldots,N$.

The symbolic program SOURC1 scans the formula (3.1) in the Gauss-Seidel algorithm according to its natural order. It produces a reduced formula which is written as a single FORTRAN statement in terms of just two symbols a and X, which are one-dimensional arrays of length n and N. The array a contains just the n nonzero elements of matrix A and the array X is the solution of equation (2.1). The array a is in a 'collapsed' one-dimensional form, the nonzero elements are arranged into an array according to its natural order, which simplifies the compilation of RESUL1 and economizes on storage. A convenient choice for the collapsed label of a nonzero element a_{ij} is I, defined as the numbers of nonzero elements encountered in processing the first X_{i+1}^{r} formulas. It can be defined as an integer-valued function of k, $k=1,2,\ldots,N^2$ by

$$I(k) = \sum_{h=1}^{k} \hat{a}_h$$

When SOURCE1 scans through formula (3.1) for a specific r, it involves those products $a_{rj}x_{i+1}^{j}$, $j=1,2,\ldots,r-1$ and $a_{rj}x_{i}^{j}$, $j=r+1, r+2, \ldots,N$. The labels are determined during the processing of the r-batch and it amounts to incrementing I by 1.

The solution of the linear equations (2.1) is shown in (3.1) which can be reduced to a simpler form. We consider a monotonically increasing sequence $\{kseq(h), h=1,\ldots,h_{max}\}$ (which may be empty), consisting of the column numbers of the $1's$ among a_{ij}, $j=1,2,\ldots,N$. Let

$$kseq(h) = I(\ K(i,j)\ ) \tag{3.2}$$

$$iseq(h) = j \tag{3.3}$$

denote the collapsed one-dimensional label of the nonzero element a_{ij}. Also, let jd be the collapsed label for the diagonal element a_{ii}.

By (3.2) and (3.3) equation (3.1) can be written in a collapsed form as

$$x(r) = -1/a_{jd} \left(- \sum_{h=1}^{h_{max}} a_{kseq(h)} * x_{iseq(h)} + b(r) \right) \qquad (3.4)$$

for $r = 1,2,\ldots,N$.

PROGRAM INPUT/OUTPUT AND EXAMPLE

The program SOURC1 is written in FORTRAN V and runs on the Univac 1108 series. It will handle matrices up to order $N = 600$. Input to the program is by cards as follows: (1) one card containing N, the order of the given matrix A, and (2) NUM, the number of nonzero elements of ith row of matrix A and $[NUM/13] + 1$ cards locating the nonzero elements by one-dimensional labeling k, $k = (i-1)*N+j$, $j = 1,2,\ldots,N$.

After reading the input data, the program enters the reduction algorithm described in Chapter III.

The main part of the output is the symbolically generated FORTRAN program RESUL1, which represents the reduced efficient Gauss-Seidel method, and is written on disk.

The input to RESUL1 consists of two one-dimensional arrays, the first array represents the numerical values of the nonzero elements of A ordered in accordance with the one-dimensional labeling mentioned above. The second array consists of the elements of b. The output of RESUL1 is the solutions of equation (2.1).

To illustrate the use of our algorithm, we give a complete description of its application by using the same example as in Chapter II.

In this case, the input to SOURC1 is

```
6
3   1   3   5
2   1   2
1   3
2   1   4
2   2   5
2   4   6
```

The program SOURC1 as written in FORTRAN V is listed in the Appendix, program 2. The generated symbolic program RESUL1 is shown in Figure 3.1.

The numerical values of the nonzero elements of A, which are part input to RESUL1, are

7.0000000+00 -3.0000000+00 -1.0000000+00 2.0000000+00 8.0000000+00

1.0000000+00 -3.0000000+00 5.0000000+00 -1.0000000+00 4.0000000+00

-2.0000000+00 6.0000000+00

and the elements of b are

-1.1000000+01 3.8000000+00 0.0000000 1.3000000+01 1.1000000+01

-2.2000000+01

After the execution of program RESUL1, we can get the solutions of equations (2.1) which are shown as

x(1) = -9.9999961-01

x(2) = 4.9999999+00

x(3) = 0.0000000

x(4) = 2.0000002+00

x(5) = 4.0000000+00

x(6) = -2.9000000+00

To illustrate the processing of our algorithm, we give a complete description of the process by using the same example as in Chapter II.

At the beginning, the program reads n, the order of matrix A and initializes $j = 0$.

Then, the program starts to simplify (3.1) by removing all product terms which contain zero elements. A set of FORTRAN statements will be generated.

As an example, we discuss the batch processing for $r = 1$. We obtain $(num, iseq(k), k = 1, num)$ by reading the input data, then formula (3.1) can be reduced into a simpler form by (3.2) and (3.3)

We find that

$$j + j+1,\ jd = j = 1,\ update\ j = j+1 = 2$$

$$iseq(1) = 3,\ kseq(1) = j = 2,\ update\ j = j+1 = 3$$

$$iseq(2) = 5,\ kseq(2) = j = 3$$

We know from (3.1)

$$x(r) = -(-B(r) + A(kseq(1))* x(iseq(1))+ A(kseq(2))*x(iseq(2)))/a(jd)$$

i.e.

$$x(1) = -(-B(1) + A(2)*x(3) + A(3)*x(5))/A(1)$$

The generated FORTRAN statements are listed in Figure 3.1.

```
      REAL*4     A(    12),B(     6),X(     6),Y(     6)
      READ(5,20) (A(K),K=1,    12)
   20 FORMAT(5E14.7)
      READ(5,20,END=99) (B(K),K=1,     6)
      DO 40 L=1,     6
      X(L)=0.
   40 Y(L)=0.
      ICOUNT=0
   50 ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.300) GO TO    70
      X(     1)=-(-B(     1)+A(     2)*X(     3)+A(     3)*X(     5)
     *)/A(     1)
      X(     2)=-(-B(     2)+A(     4)*X(     1)
     *)/A(     5)
      X(     3)=B(     3)/A(     6)
      X(     4)=-(-B(     4)+A(     7)*X(     1)
     *)/A(     8)
      X(     5)=-(-B(     5)+A(     9)*X(     2)
     *)/A(    10)
      X(     6)=-(-B(     6)+A(    11)*X(     4)
     *)/A(    12)


      DO 60 L=1,     6
      TEMP=ABS(Y(L)-X(L))
      IF(TEMP.GT.1E-04) GO TO 80
   60 CONTINUE
   70 DO 75 I=1,     6
      WRITE(6,85) I,X(I)
   85 FORMAT(25X,'X(',I5,')=',E14.7)
   75 CONTINUE
   25 FORMAT(20X,' THE ITERATIVE METHOD FAILED')
      STOP
   80 DO 90 L=1,     6
      Y(L)=X(L)
   90 CONTINUE
      GO TO 50
   99 END
```

Figure 3.1

DISCUSSION

This paper has shown that the efficient Gauss-Seidel method is applicable to a given system if the existence of convergence is known and the diagonal of matrix A contains no zero elements. If it does have zero elements but A is nonsingular, it is always possible to get a non-singular matrix by permuting rows and columns. But there is no guarantee that the approximation of the permuted matrix will converge to the solution, unless the matrix can satisfy theorem 3.1 or 3.2. If the rate of convergence is known to be rapid, then this method can be valuable.

Theorem: 3.1 With A written in the form $A = L + D + U$, which can be solved for x_{i+1} to give

$$x_{i+1} = -(D + L)^{-1} U x_i + (D + L)^{-1} b$$

Let $B = -(D + L)^{-1} U$, a sufficient condition for convergence is then that $\| (D + L)^{-1} U \| < 1$, where $\| \quad \|$ is the spectral radius of B (see 1, p. 432).

Theorem: 3.2 If the matrix A is both symmetric and positive definite, the Gauss-Seidel iteration converges independently of the initial vector (see 1, p. 432).

The entire program SOURC1 requires only $2N$ words for storing the key information to locate the nonzero elements and the symbolically generated program RESUL1 stores only the nonzero elements. This compares to the total need of core allocation of the Gauss-Seidel algorithm which is at least N^2 words. Thus makes quite a difference if the order N of the matrix is larger than 100. Figure 3.3 shows the comparison of core allocation.

Another major consideration is the execution time. In order to test the execution time of our algorithm, we use the same examples as in Chapter II. A set of results are listed in Table 3.1 which verifies that the reduced method is much more efficient. Figure 3.2 shows the comparison of execution time.

Table 3.1 Comparison of Execution Time Between Gauss-Seidel Algorithm and Reduced Gauss-Seidel

Algorithm / CPU Time (Sec) / Order of Matrices	Gauss-Seidel	Reduced Gauss-Seidel	RESULT
10	.190	1.864	.106
25	.913	2.769	.149
50	3.682	4.198	.317
75	6.643	5.101	.459
100	14.540	6.655	.669
125	18.918	8.178	.737
150	29.101	8.780	.921
175	37.125	10.501	1.048
200	48.186	11.769	1.370
225	59.359	13.957	1.570

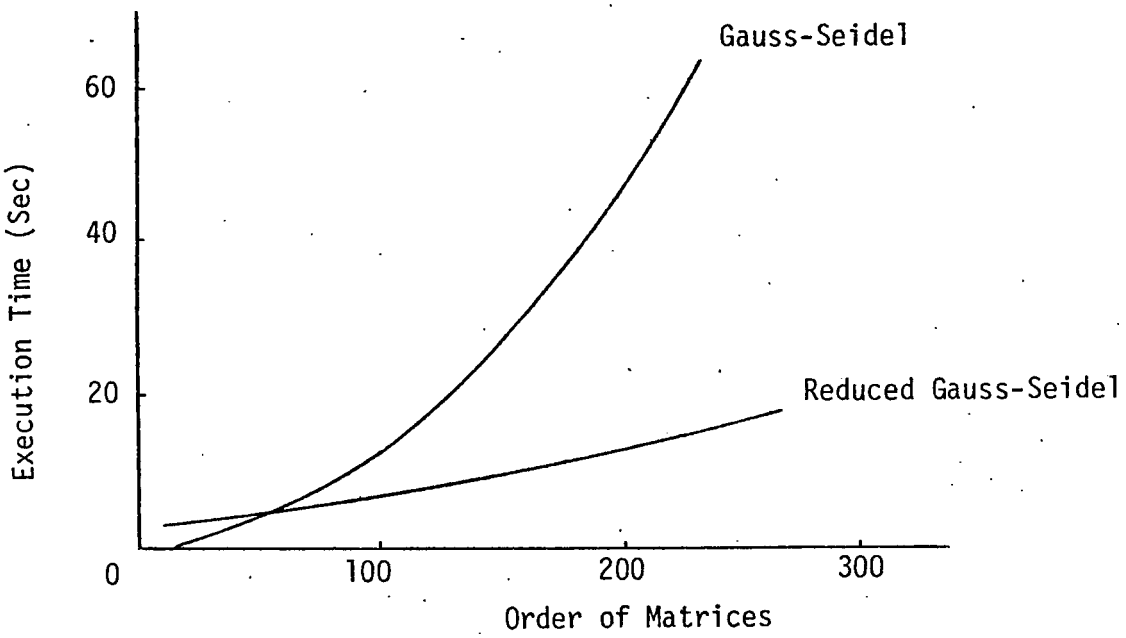


Figure 3.2

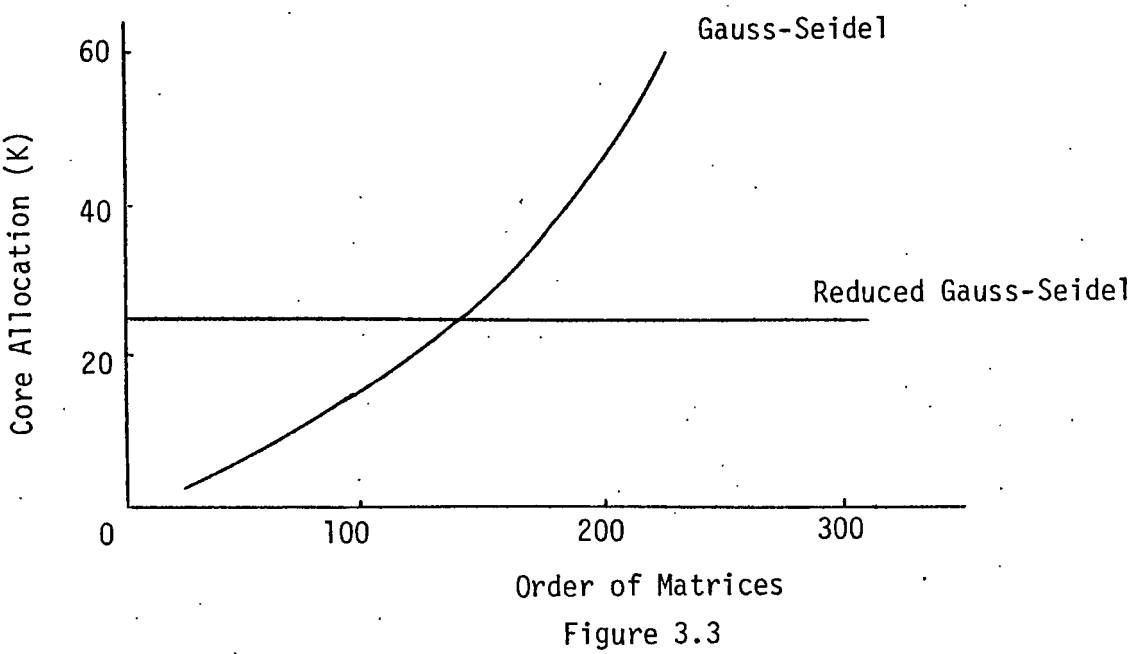


Figure 3.3

COMPARISON OF TWO METHODS

It should be emphasized that there are no hard-and-fast rules in choosing between direct method and iterative method. Usually much depends on the form of the matrix, the rate of convergence, and so on. For most matrices, iterative method requires more computation to achieve a desired degree of convergence than the direct method. Iterative methods are used mainly in those problems for which convergence is known to be rapid, so that the solution is obtained with much less work than that of a direct method. Iterative methods are particularly advantageous for the very large matrices that often occur in numerical solution of partial differential equations.

The efficiency of an algorithm will be judged by two main criteria besides the accuracy of the solution: (1) How fast is it? (2) How much core does it use?

These two criteria are aimed at the evaluation of algorithms for the solution of high-order systems (100 equations or more) on a digital computer. Because of the formidable amount of computation and core allocation requires to solve (2.1) for large systems, the need to answer the questions is clear.

A set of examples have been tested on both methods to solve an unsteady-state heat flow problem with fixed boundary condition. The results, including core allocation and execution time, show that the Gauss-Seidel method may compare favorably with the Crout method. In general, this may not be true, because the density of these matrices is low and the rate of convergence is very fast.

BIBLIOGRAPHY

1. Fox, L., A First Course in Numerical Analysis, Oxford University Press, N.Y., 1964.

2. Ralston, A., An Introduction to Numerical Linear Algebra, McGraw-Hill, N.Y., 1965.

3. Gustavson, F. G., Symbolic Generation of an Optimal Crout Algorithm, Journal of the Association for Computing Machinery, Vol. 17, No. 1, January 1970, pages 87-109.

4. Pooch, V. W., Indexing Techniques for Sparse Matrices, Computing Surveys, Vol. 5, No. 2, June 1973, pages 109-133.

5. Gerald, C. F., Applied Numerical Analysis, Addison-Wesley Publishing Company, 1970.

6. Varga, R. S., Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1962.

APPENDIX

PROGRAM I

SOURCE

```
C       MAIN PROGRAM
C       AFTER READING THE INPUT DATA AND INITIALIZING THE BIT MAP LIST,
C       THE PROGRAM ENTERS THE REDUCTION ALGORITHM, WHICH IS BASED ON THE
C       CROUT METHOD, AS DETAILED BELOW
        IMPLICIT INTEGER (B,C,E,R)
        COMMON /AREA1/IPOINT,IWORD(10000)
        COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
       *B(600),BCR(600),BCC(600)
        DIMENSION ICARD(12)
        DIMENSION INPUT(16)
        CALL CPUTIM(IVAL1)
        COMMON /AREA3/ECR(600),ECC(600)
        IPOINT=1
        I=0
        J=0
        K=0
        DO 50 KK=1,10000
        IWORD(KK)=0
   50   CONTINUE
        IPOINT=1
        READ(5,10) N
   10   FORMAT(2I5)
    3   READ(5,20,END=21) (INPUT(II),II=1,16)
   20   FORMAT(16I5)
C       INITIALIZING THE BIT MAP LIST
        DO 30 II=1,16
        CALL STORE(INPUT(II),I)
   30   CONTINUE
        GO TO 3
   21   CONTINUE
C       ENTER THE REDUCTION ALGORITHM
        DO 40 M=1,N
        CALL CHART1
        CALL CHART2
        CALL CHART3
        CALL CHART4
        IF(M.EQ.N) GO TO 99
        CALL CHART5
        CALL CHART6
        CALL CHART7
   40   CONTINUE
   99   CONTINUE
        ENDFILE 2
        REWIND 2
        WRITE(3,400) I,N,J,N,N
  400   FORMAT(T7,'REAL*4     A(',I5,'),B(',I5,'),C(',I5,'),Y(',I5,'),X(',
       *I5,')')
        WRITE(3,410) I
  410   FORMAT(T5,'5 READ(5,10,END=99) (A(I),I=1,',I5,')')
        WRITE(3,420)
  420   FORMAT(T4,'10 FORMAT(1P5E14.7)')
        WRITE(3,450) N
  450   FORMAT(T7,'READ (5,10) (B(I),I=1,',I5,')')
   98   READ(2,60,END=999) (ICARD(II),II=1,12)
   60   FORMAT(12A6)
        WRITE(3,60) (ICARD(II),II=1,12)
        GO TO 98
  999   CALL CHART8
```

```
      WRITE(3,430)
  430 FORMAT(T7,'WRITE(6,40)')
      WRITE(3,440)
  440 FORMAT(T4,'40 FORMAT(5X,''THE SOLUTIONS OF EQUATIONS   AX = B  ''
     *,////)')
      WRITE(3,470) N
  470 FORMAT(T7,'DO 30 I=1,',I5)
      WRITE(3,471)
  471 FORMAT(T7,'WRITE(6,20) I,X(I)')
      WRITE(3,472)
  472 FORMAT(T4,'20 FORMAT(10X,''X('',I5,'')='',1PE14.7)')
      WRITE(3,473)
  473 FORMAT(T4,'30 CONTINUE')
      WRITE(3,500)
  500 FORMAT(T7,'GO TO 5')
      WRITE(3,510)
  510 FORMAT(T4,'99 END')
      CALL CPUTIM(IVAL2)
      IE=(IVAL2-IVAL1)*200
      WRITE(6,66) IVAL1,IVAL2,IE
   66 FORMAT(1H ,///,' THE CPU TIME IS ',3I10)
      ENDFILE 3
      STOP
      END



      SUBROUTINE STORE(K,IBOOL)
C     STORE THE INFORMATION INTO THE BIT MAP.
      COMMON /AREA1/IPOINT,IWORD(10000)
      KK=K-1
      I=KK/36+1
      J=MOD(KK,36)
      FLD(J,1,IWORD(I))=IBOOL
      RETURN
      END



      INTEGER FUNCTION IBIT(K)
C     INITIALIZING THE BIT MAP.
      COMMON /AREA1/IPOINT,IWORD(10000)
      KK=K-1
      I=KK/36+1
      J=MOD(KK,36)
      IBIT=FLD(J,1,IWORD(I))
      RETURN
      END



      SUBROUTINE GET(JBIT)
C     RETRIEVE THE INFORMATION FROM THE BIT MAP.
      COMMON /AREA1/IPOINT,IWORD(10000)
      DATA I/-1/
      I=I+1
      IF(I.LE.35) GO TO 10
      I=0
      IPOINT=IPOINT+1
   10 JBIT=FLD(I,1,IWORD(IPOINT))
      RETURN
      ENTRY PUT
```

```
C       STORE THE INFORMATION INTO THE BIT MAP.
        FLD(I,1,IWORD(IPOINT))=1
        RETURN
        END



        SUBROUTINE CHART1
C       DETERMINE THE SEQUENCES BCC,B,AND BCR, SIMULTANEOUSLY UPDATING
C       CC((V),V=1,2,....,M-1 FOR THE MTH BATCHES.
        IMPLICIT INTEGER (B,C,E,R)
        COMMON /AREA1/IPOINT,IWORD(10000)
        COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
       *B(600),BCR(600),BCC(600)
        K=K+1
        IV=1
        IC=0
        MM=M-1
        IF(M.EQ.1) GO TO 30
   10   IF(IBIT(C(IV)).EQ.0) GO TO 20
        CC(IV)=CC(IV)+1
        IC=IC+1
        BCC(IC)=CC(IV)
        B(IC)=R(IV)
        BCR(IC)=RC(IV)
   20   IF(IV.EQ.MM) GO TO 30
        IV=IV+1
        GO TO 10
   30   MAXIC=IC
        RETURN
        END



        SUBROUTINE CHART2
C       DETERMINE R(M),RC(M), AND JD (JD IS THE COLLAPSED FORMULA LABEL
C       OF THE DIAGONAL ELEMENT L(M,M)).
        IMPLICIT INTEGER (B,C,E,R)
        COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
       *B(600),BCR(600),BCC(600)
        R(M)=K+1
        RC(M)=J+1
        JD=J+1
        RETURN
        END



        SUBROUTINE CHART3
C       PROCESS THE L-FORMULAS AND WRITES OUT A REDUCED CROUT FORMULA IN A
C       COLLAPSED FORM
        IMPLICIT INTEGER (B,C,E,R)
        COMMON /AREA1/IPOINT,IWORD(10000)
        COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
       *B(600),BCR(600),BCC(600)
        COMMON /AREA3/ECR(600),ECC(600)
        IF(MAXIC.EQ.0) GO TO 80
        IT=1
   10   IC=1
        ID=0
   20   IF(IBIT(B(IC)+IT-1).EQ.0) GO TO 30
        BCR(IC)=BCR(IC)+1
```

```
      ID=ID+1
      ECR(ID)=BCR(IC)
      ECC(ID)=BCC(IC)
   30 IF(MAXIC.EQ.IC) GO TO 40
      IC=IC+1
      GO TO 20
   40 CALL GET(JBIT)
      IF(JBIT.EQ.0) GO TO 60
      IF(ID.EQ.0) GO TO 50
      I=I+1
      J=J+1
C     WRITE F3
      CALL PRINT2(ID)
      ID=0
      GO TO 70
   50 I=I+1
      J=J+1
C     WRITE F1
      WRITE(2,350) J,I
  350 FORMAT(1H ,T7,'C(',I5,')=A(',I5,')')
      GO TO 70
   60 IF(ID.EQ.0) GO TO 70
      J=J+1
      CALL PUT
C     WRITE F2
      CALL PRINT1(ID)
      ID=0
   70 IF(IT.EQ.(N-M+1)) RETURN
      K=K+1
      IT=IT+1
      GO TO 10
   80 CALL GET(JBIT)
      IF(JBIT.EQ.0) GO TO 90
      I=I+1
      J=J+1
C     WRITE F1
      WRITE(2,350) J,I
   90 IF(K.EQ.((M-1)*(2*N-M)+N)) RETURN
      K=K+1
      GO TO 80
      END



      SUBROUTINE CHART4
C     DETERMINE THE SEQUENCES BCR,B, AND BCC, SIMULTANEOUSLY UPDATING
C     RC(V),V=1,2,...,M-1 FOR THE MTH BATCH AND WRITE OUT THE
C     FORMULA FOR Y(M).
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA1/IPOINT,IWORD(10000)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA LC,LY,LP/'-C(',')*Y(',')'/
      K=K+1
      IV=1
      IC=0
      MM=M-1
      IF(M.EQ.1) GO TO 30
   10 IF(IBIT(R(IV)).EQ.0) GO TO 20
      RC(IV)=RC(IV)+1
      IC=IC+1
```

```
      BCR(IC)=RC(IV)
      B(IC)=C(IV)
      BCC(IC)=CC(IV)
      ECR(IC)=RC(IV)
      ECC(IC)=IV
   20 IF(IV.EQ.MM) GO TO 30
      IV=IV+1
      GO TO 10
   30 MAXIC=IC
      IF(IC.NE.0) GO TO 40
C     WRITE B1
      WRITE(2,300) M,M,JD
  300 FORMAT(T7,'Y(',I5,')=B(',I5,')/C(',I5,')')
      RETURN
C     WRITE B2
   40 IP=IC-3
      IF(IP.LE.0) GO TO 50
      WRITE(2,310) M,(LC,ECR(II),LY,ECC(II),LP,II=1,3)
  310 FORMAT(T7,'Y(',I5,')=(',3(A3,I5,A4,I5,A1))
      WRITE(2,320) (LC,ECR(II),LY,ECC(II),LP,II=4,IC)
  320 FORMAT((T6,'+',3(A3,I5,A4,I5,A1)))
      GO TO 60
   50 WRITE(2,310) M,(LC,ECR(II),LY,ECC(II),LP,II=1,IC)
   60 WRITE(2,350) M,JD
  350 FORMAT(T6,'+','+B(',I5,'))/C(',I5,')')
      RETURN
      END


      SUBROUTINE CHART5
C     DETERMINE C(M) AND CC(M).
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      C(M)=K
      CC(M)=J
      RETURN
      END


      SUBROUTINE CHART6
C     PROCESS THE U-FORMULA AND WRITE A REDUCED CROUT FORMULA IN A
C     COLLAPSED FORM.
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA1/IPOINT,IWORD(10000)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      IF(MAXIC.EQ.0) GO TO 80
      IT=1
   10 IC=1
      ID=0
   20 IF(IBIT(B(IC)+IT).EQ.0) GO TO 30
      BCC(IC)=BCC(IC)+1
      ID=ID+1
      ECR(ID)=BCR(IC)
      ECC(ID)=BCC(IC)
   30 IF(MAXIC.EQ.IC) GO TO 40
      IC=IC+1
      GO TO 20
   40 CALL GET(JBIT)
```

```
      IF(JBIT.EQ.0) GO TO 60
      IF(ID.EQ.0) GO TO 50
      I=I+1
      J=J+1
C     WRITE F3
      CALL PRINT4(ID)
      ID=0
      GO TO 70
   50 I=I+1
      J=J+1
C     WRITE F1
      WRITE(2,300) J,I,JD
  300 FORMAT(T7,'C(',I5,')=A(',I5,')/C(',I5,')')
      GO TO 70
   60 IF(ID.EQ.0) GO TO 70
      J=J+1
C     WRITE F2
      CALL PRINT3(ID)
      CALL PUT
      ID=0
   70 IF(IT.EQ.(N-M)) RETURN
      K=K+1
      IT=IT+1
      GO TO 10
   80 CALL GET(JBIT)
      IF(JBIT.EQ.0) GO TO 90
      I=I+1
      J=J+1
C     WRITE F1
      WRITE(2,300) J,I,JD
   90 IF(K.EQ.(M*(2*N-M))) RETURN
      K=K+1
      GO TO 80
      END



      SUBROUTINE CHART7
C     UPDATE THE SEQUENCES C(V) AND R(V), V=1,2,...,M-1
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      MM=M-1
      IF(MM.LE.0) RETURN
      DO 10 IV=1,MM
      C(IV)=C(IV)+1
      R(IV)=R(IV)+1
   10 CONTINUE
      RETURN
      END



      SUBROUTINE PRINT4(ID)
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA I1,I2,I3,I4,I5,I6,I7/'-C(','C(',')C(',')',')=(',')/C(','A('/
      IP=ID-2
      IF(IP.LE.0) GO TO 10
      WRITE(2,100) I2,J,I5,I7,I,I4,(I1,ECR(II),I3,ECC(II),I4,II=1,2)
```

```
      WRITE(2,110) (I1,ECR(II),I3,ECC(II),I4,II=3,ID)
  110 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
      GO TO 20
   10 WRITE(2,100) I2,J,I5,I7,I,I4,(I1,ECR(II),I3,ECC(II),I4,II=1,ID)
  100 FORMAT(T7,A2,I5,A3,A2,I5,A1,2(A3,I5,A4,I5,A1))
   20 WRITE(2,120) I6,JD,I4
  120 FORMAT(T6,'*',A4,I5,A1)
      RETURN
      END



      SUBROUTINE PRINT3(ID)
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA I1,I2,I3,I4,I5,I6,I7/'-C(','C(',')*C(',')',')=(',')/C(','A('/
      IP=ID-3
      IF(IP.LE.0) GO TO 10
      WRITE(2,100) I2,J,I5,(I1,ECR(II),I3,ECC(II),I4,II=1,3)
  100 FORMAT(T7,A2,I5,A3,3(A3,I5,A4,I5,A1))
      WRITE(2,110) (I1,ECR(II),I3,ECC(II),I4,II=4,ID)
  110 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
      GO TO 20
   10 WRITE(2,100) I2,J,I5,(I1,ECR(II),I3,ECC(II),I4,II=1,ID)
   20 WRITE(2,120) I6,JD,I4
  120 FORMAT(T6,'*',A4,I5,A1)
      RETURN
      END



      SUBROUITNE PRINT2(ID)
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA I1,I2,I3,I4,I5,I6/'C(',')*C(',')','-C(','A(',')='/
      IP=ID-2
      IF(IP.LE.0) GO TO 10
      WRITE(2,100) I1,J,I6,I5,I,I3,(I4,ECR(II),I2,ECC(II),I3,II=1,2)
  100 FORMAT(T7,A2,I5,2A2,I5,A1,2(A3,I5,A4,I5,A1))
      WRITE(2,110) (I4,ECR(II),I2,ECC(II),I3,II=3,ID)
  110 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
      GO TO 20
   10 WRITE(2,100) I1,J,I6,I5,I,I3,(I4,ECR(II),I2,ECC(II),I3,II=1,ID)
   20 RETURN
      END



      SUBROUTINE PRINT1(ID)
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA I1,I2,I3,I4,I5,I6/'C(',')*C(',')','-C(','A(',')='/
      IP=ID-3
      IF(IP.LE.0) GO TO 10
      WRITE(2,100) I1,J,I6,(I4,ECR(II),I2,ECC(II),I3,II=1,3)
  100 FORMAT(T7,A2,I5,A2,3(A3,I5,A4,I5,A1))
```

```
      WRITE(2,110) (I4,ECR(II),I2,ECC(II),I3,II=4,ID)
  110 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
      GO TO 20
   10 WRITE(2,100) I1,J,I6,(I4,ECR(II),I2,ECC(II),I3,II=1,ID)
   20 RETURN
      END


      SUBROUTINE CHART8
C     WRITE OUT THE REDUCED FORMULA FOR SOLUTION X(I),I=1,N
      IMPLICIT INTEGER (B,C,E,R)
      COMMON /AREA2/N,M,I,J,K,JD,MAXIC,C(600),R(600),CC(600),RC(600),
     *B(600),BCR(600),BCC(600)
      COMMON /AREA3/ECR(600),ECC(600)
      DATA LC,LX,LP/'-C(',')*X(',')'/
      WRITE(3,300) M,M
  300 FORMAT(T7,'X(',I5,')=Y(',I5,')')
      MM=M-1
      DO 100 L=MM,1,-1
      IV=N
      ID=0
   10 IF(IV.LE.L) GO TO 30
      IF(IBIT(C(L)).EQ.0) GO TO 20
      ID=ID+1
      ECR(ID)=CC(L)
      ECC(ID)=IV
      CC(L)=CC(L)-1
   20 IV=IV-1
      C(L)=C(L)-1
      GO TO 10
   30 IF(ID.NE.0) GO TO 40
      WRITE(3,300) L,L
      GO TO 100
   40 WRITE(3,310) L,L,ECR(1),ECC(1)
  310 FORMAT(T7,'X(',I5,')=Y(',I5,')','-C(',I5,')*X(',I5,')')
      IF((ID-1).LE.0) GO TO 100
      WRITE(3,320) (LC,ECR(II),LX,ECC(II),LP,II=2,ID)
  320 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
  100 CONTINUE
      RETURN
      END
```

OUTPUT OF PROGRAM SOURCE

SYMBOLIC GENERATED PROGRAM

RESULT

```
      REAL*8     A(  281),B(   57),C(  446),Y(   57),X(   57)
    5 READ(5,10,END=99) (A(I),I=1,  281)
   10 FORMAT(1PSE14.7)
      READ (5,10) (B(I),I=1,   57)
      C(    1)=A(    1)
      C(    2)=A(    2)
      C(    3)=A(    3)
      C(    4)=A(    4)
      C(    5)=A(    5)
      C(    6)=A(    6)
      Y(    1)=B(    1)/C(    1)
      C(    7)=A(    7)/C(    1)
      C(    8)=A(    8)/C(    1)
      C(    9)=A(    9)/C(    1)
      C(   10)=A(   10)/C(    1)
      C(   11)=A(   11)/C(    1)
      C(   12)=A(   12)/C(    1)
      C(   13)=A(   13)/C(    1)
      C(   14)=A(   14)/C(    1)
      C(   15)=A(   15)/C(    1)
      C(   16)=A(   16)-C(    2)*C(    7)
      C(   17)=-C(    3)*C(    7)
      C(   18)=-C(    4)*C(    7)
      C(   19)=A(   17)
      C(   20)=A(   18)
      C(   21)=-C(    5)*C(    7)
      C(   22)=-C(    6)*C(    7)
      Y(    2)=(-C(    2)*Y(    1)
     *+B(    2))/C(   16)
      C(   23)=(A(   19)-C(    2)*C(    8)
     *)/C(   16)
      C(   24)=(A(   20)-C(    2)*C(    9)
     *)/C(   16)
      C(   25)=(A(   21)-C(    2)*C(   10)
     *)/C(   16)
      C(   26)=(A(   22)-C(    2)*C(   11)
     *)/C(   16)
      C(   27)=(A(   23)-C(    2)*C(   12)
     *)/C(   16)
      C(   28)=(A(   24)-C(    2)*C(   13)
     *)/C(   16)
      C(   29)=(A(   25)-C(    2)*C(   14)
     *)/C(   16)
      C(   30)=(A(   26)-C(    2)*C(   15)
     *)/C(   16)
      C(   31)=A(   27)
      C(   32)=A(   28)
      C(   33)=A(   29)
      Y(    3)=B(    3)/C(   31)
      C(   34)=A(   30)/C(   31)
      C(   35)=A(   31)/C(   31)
      C(   36)=A(   32)-C(   32)*C(   34)
      C(   37)=-C(   33)*C(   34)
      Y(    4)=(-C(   32)*Y(    3)
     *+B(    4))/C(   36)
      C(   38)=(-C(   32)*C(   35)
     *)/C(   36)
      C(   39)=A(   33)
```

```
      C(    40)=A(    34)
      C(    41)=A(    35)
      Y(     5)=B(     5)/C(    39)
      C(    42)=A(    36)/C(    39)
      C(    43)=A(    37)/C(    39)
      C(    44)=A(    38)/C(    39)
      C(    45)=A(    39)/C(    39)
      C(    46)=A(    40)-C(    40)*C(    42)
      C(    47)=A(    41)
      C(    48)=-C(    41)*C(    42)
      Y(    .6)=(-C(    40)*Y(     5)
     *+B(     6))/C(    46)
      C(    49)=A(    42)/C(    46)
      C(    50)=(A(    43)-C(    40)*C(    43)
     *)/C(    46)
      C(    51)=(A(    44)-C(    40)*C(    44)
     *)/C(    46)
      C(    52)=(A(    45)-C(    40)*C(    45)
     *)/C(    46)
      C(    53)=A(    46)-C(    47)*C(    49)
      C(    54)=-C(    48)*C(    49)
      Y(     7)=(-C(    47)*Y(     6)
     *+B(     7))/C(    53)
      C(    55)=(-C(    47)*C(    50)
     *)/C(    53)
      C(    56)=(-C(    47)*C(    51)
     *)/C(    53)
      C(    57)=(-C(    47)*C(    52)
     *)/C(    53)
      C(    58)=A(    47)-C(     3)*C(     8)-C(    17)*C(    23)
      C(    59)=A(    48)-C(     4)*C(     8)-C(    18)*C(    23)
      C(    60)=-C(    19)*C(    23)
      C(    61)=-C(    20)*C(    23)
      C(    62)=A(    49)
      C(    63)=-C(     5)*C(     8)-C(    21)*C(    23)
      C(    64)=-C(     6)*C(     8)-C(    22)*C(    23)
      Y(     8)=(-C(     3)*Y(     1)-C(    17)*Y(     2)
     *+B(     8))/C(    58)
      C(    65)=(A(    50)-C(     3)*C(     9)-C(    17)*C(    24)
     *)/C(    58)
      C(    66)=(-C(     3)*C(    10)-C(    17)*C(    25)
     *)/C(    58)
      C(    67)=(-C(     3)*C(    11)-C(    17)*C(    26)
     *)/C(    58)
      C(    68)=(A(    51)-C(     3)*C(    12)-C(    17)*C(    27)
     *)/C(    58)
      C(    69)=(-C(     3)*C(    13)-C(    17)*C(    28)
     *)/C(    58)
      C(    70)=(-C(     3)*C(    14)-C(    17)*C(    29)
     *)/C(    58)
      C(    71)=(-C(     3)*C(    15)-C(    17)*C(    30)
     *)/C(    58)
      C(    72)=A(    52)-C(     4)*C(     9)-C(    18)*C(    24)
     *-C(    59)*C(    65)
      C(    73)=-C(    19)*C(    24)-C(    60)*C(    65)
      C(    74)=-C(    20)*C(    24)-C(    61)*C(    65)
      C(    75)=-C(    62)*C(    65)
      C(    76)=-C(     5)*C(     9)-C(    21)*C(    24)-C(    63)*C(    65)
      C(    77)=A(    53)
      C(    78)=-C(     6)*C(     9)-C(    22)*C(    24)-C(    64)*C(    65)
      Y(     9)=(-C(     4)*Y(     1)-C(    18)*Y(     2)-C(    59)*Y(     8)
     *+B(     9))/C(    72)
```

```
 C(   79)=(-C(    4)*C(   10)-C(   18)*C(   25)-C(   59)*C(   66)
*)/C(   72)
 C(   80)=(-C(    4)*C(   11)-C(   18)*C(   26)-C(   59)*C(   67)
*)/C(   72)
 C(   81)=(-C(    4)*C(   12)-C(   18)*C(   27)-C(   59)*C(   68)
*)/C(   72)
 C(   82)=(-C(    4)*C(   13)-C(   18)*C(   28)-C(   59)*C(   69)
*)/C(   72)
 C(   83)=(-C(    4)*C(   14)-C(   18)*C(   29)-C(   59)*C(   70)
*)/C(   72)
 C(   84)=(-C(    4)*C(   15)-C(   18)*C(   30)-C(   59)*C(   71)
*)/C(   72)
 C(   85)=A(   54)
 C(   86)=A(   55)
 Y(   10)=B(   10)/C(   85)
 C(   87)=A(   56)/C(   85)
 C(   88)=A(   57)/C(   85)
 C(   89)=A(   58)-C(   19)*C(   25)-C(   60)*C(   66)
*-C(   73)*C(   79)-C(   86)*C(   87)
 C(   90)=A(   59)-C(   41)*C(   43)-C(   48)*C(   50)
*-C(   54)*C(   55)
 C(   91)=A(   60)
 C(   92)=-C(   20)*C(   25)-C(   61)*C(   66)-C(   74)*C(   79)
 C(   93)=-C(   62)*C(   66)-C(   75)*C(   79)
 C(   94)=-C(    5)*C(   10)-C(   21)*C(   25)-C(   63)*C(   66)
*-C(   76)*C(   79)
 C(   95)=-C(   77)*C(   79)
 C(   96)=-C(    6)*C(   10)-C(   22)*C(   25)-C(   64)*C(   66)
*-C(   78)*C(   79)
 Y(   11)=(-C(   19)*Y(    2)-C(   60)*Y(    8)-C(   73)*Y(    9)
*-C(   86)*Y(   10)
*+B(   11))/C(   89)
 C(   97)=(A(   61)-C(   19)*C(   26)-C(   60)*C(   67)
*-C(   73)*C(   80)-C(   86)*C(   88)
*)/C(   89)
 C(   98)=(-C(   19)*C(   27)-C(   60)*C(   68)-C(   73)*C(   81)
*)/C(   89)
 C(   99)=(-C(   19)*C(   28)-C(   60)*C(   69)-C(   73)*C(   82)
*)/C(   89)
 C(  100)=(A(   62)-C(   19)*C(   29)-C(   60)*C(   70)
*-C(   73)*C(   83)
*)/C(   89)
 C(  101)=(-C(   19)*C(   30)-C(   60)*C(   71)-C(   73)*C(   84)
*)/C(   89)
 C(  102)=A(   63)-C(   41)*C(   44)-C(   48)*C(   51)
*-C(   54)*C(   56)-C(   90)*C(   97)
 C(  103)=A(   64)-C(   91)*C(   97)
 C(  104)=-C(   20)*C(   26)-C(   61)*C(   67)-C(   74)*C(   80)
*-C(   92)*C(   97)
 C(  105)=-C(   62)*C(   67)-C(   75)*C(   80)-C(   93)*C(   97)
 C(  106)=-C(    5)*C(   11)-C(   21)*C(   26)-C(   63)*C(   67)
*-C(   76)*C(   80)-C(   94)*C(   97)
 C(  107)=-C(   77)*C(   80)-C(   95)*C(   97)
 C(  108)=-C(    6)*C(   11)-C(   22)*C(   26)-C(   64)*C(   67)
*-C(   78)*C(   80)-C(   96)*C(   97)
 C(  109)=A(   65)
 Y(   12)=(-C(   41)*Y(    5)-C(   48)*Y(    6)-C(   54)*Y(    7)
*-C(   90)*Y(   11)
*+B(   12))/C(  102)
 C(  110)=A(   66)/C(  102)
 C(  111)=(-C(   90)*C(   98)
*)/C(  102)
```

```
*      C(  112)=(-C(   90)*C(   99)
*     *)/C(  102)
*      C(  113)=(-C(   90)*C(  100)
*     *)/C(  102)
*      C(  114)=(-C(   90)*C(  101)
*     *)/C(  102)
*      C(  115)=(A(   67)-C(   41)*C(   45)-C(   48)*C(   52)
*     *-C(   54)*C(   57)
*     *)/C(  102)
*      C(  116)=A(   68)-C(  103)*C(  110)
*      C(  117)=-C(  104)*C(  110)
*      C(  118)=-C(  105)*C(  110)
*      C(  119)=-C(  106)*C(  110)
*      C(  120)=-C(  107)*C(  110)
*      C(  121)=-C(  108)*C(  110)
*      C(  122)=-C(  109)*C(  110)
*      Y(   13)=(-C(   91)*Y(   11)-C(  103)*Y(   12)
*     *+B(   13))/C(  116)
*      C(  123)=(-C(   91)*C(   98)-C(  103)*C(  111)
*     *)/C(  116)
*      C(  124)=(-C(   91)*C(   99)-C(  103)*C(  112)
*     *)/C(  116)
*      C(  125)=(-C(   91)*C(  100)-C(  103)*C(  113)
*     *)/C(  116)
*      C(  126)=(-C(   91)*C(  101)-C(  103)*C(  114)
*     *)/C(  116)
*      C(  127)=(-C(  103)*C(  115)
*     *)/C(  116)
*      C(  128)=A(   69)-C(   33)*C(   35)-C(   37)*C(   38)
*      Y(   14)=(-C(   20)*Y(    2)-C(   33)*Y(    3)-C(   37)*Y(    4)
*     *-C(   61)*Y(    8)-C(   74)*Y(    9)-C(   92)*Y(   11)
*     *-C(  104)*Y(   12)-C(  117)*Y(   13)
*     *+B(   14))/C(  128)
*      C(  129)=(-C(   20)*C(   27)-C(   61)*C(   68)-C(   74)*C(   81)
*     *-C(   92)*C(   98)-C(  104)*C(  111)-C(  117)*C(  123)
*     *)/C(  128)
*      C(  130)=(-C(   20)*C(   28)-C(   61)*C(   69)-C(   74)*C(   82)
*     *-C(   92)*C(   99)-C(  104)*C(  112)-C(  117)*C(  124)
*     *)/C(  128)
*      C(  131)=(-C(   20)*C(   29)-C(   61)*C(   70)-C(   74)*C(   83)
*     *-C(   92)*C(  100)-C(  104)*C(  113)-C(  117)*C(  125)
*     *)/C(  128)
*      C(  132)=(-C(   20)*C(   30)-C(   61)*C(   71)-C(   74)*C(   84)
*     *-C(   92)*C(  101)-C(  104)*C(  114)-C(  117)*C(  126)
*     *)/C(  128)
*      C(  133)=(-C(  104)*C(  115)-C(  117)*C(  127)
*     *)/C(  128)
*      C(  134)=A(   70)-C(   62)*C(   68)-C(   75)*C(   81)
*     *-C(   93)*C(   98)-C(  105)*C(  111)-C(  118)*C(  123)
*      C(  135)=-C(    5)*C(   12)-C(   21)*C(   27)-C(   63)*C(   68)
*     *-C(   76)*C(   81)-C(   94)*C(   98)-C(  106)*C(  111)
*     *-C(  119)*C(  123)
*      C(  136)=-C(   77)*C(   81)-C(   95)*C(   98)-C(  107)*C(  111)
*     *-C(  120)*C(  123)
*      C(  137)=-C(    6)*C(   12)-C(   22)*C(   27)-C(   64)*C(   68)
*     *-C(   78)*C(   81)-C(   96)*C(   98)-C(  108)*C(  111)
*     *-C(  121)*C(  123)
*      C(  138)=-C(  109)*C(  111)-C(  122)*C(  123)
*      Y(   15)=(-C(   62)*Y(    8)-C(   75)*Y(    9)-C(   93)*Y(   11)
*     *-C(  105)*Y(   12)-C(  118)*Y(   13)
*     *+B(   15))/C(  134)
*      C(  139)=(-C(   62)*C(   69)-C(   75)*C(   82)-C(   93)*C(   99)
```

```
*-C(  105)*C(  112)-C(  118)*C(  124)
*)/C(  134)
 C(  140)=(-C(   62)*C(   70)-C(   75)*C(   83)-C(   93)*C(  100)
*-C(  105)*C(  113)-C(  118)*C(  125)
*)/C(  134)
 C(  141)=(-C(   62)*C(   71)-C(   75)*C(   84)-C(   93)*C(  101)
*-C(  105)*C(  114)-C(  118)*C(  126)
*)/C(  134)
 C(  142)=(-C(  105)*C(  115)-C(  118)*C(  127)
*)/C(  134)
 C(  143)=A(   71)
 C(  144)=A(   72)
 C(  145)=A(   73)
 Y(   16)=B(   16)/C(  143)
 C(  146)=A(   74)/C(  143)
 C(  147)=A(   75)/C(  143)
 C(  148)=A(   76)
 C(  149)=A(   77)
 Y(   17)=B(   17)/C(  148)
 C(  150)=A(   78)/C(  148)
 C(  151)=A(   79)-C(  144)*C(  146)
 C(  152)=-C(  145)*C(  146)
 C(  153)=A(   80)
 C(  154)=A(   81)
 Y(   18)=(-C(  144)*Y(   16)
*+B(   18))/C(  151)
 C(  155)=(-C(  144)*C(  147)
*)/C(  151)
 C(  156)=A(   82)/C(  151)
 C(  157)=A(   83)/C(  151)
 C(  158)=A(   84)
 C(  159)=A(   85)
 C(  160)=A(   86)
 C(  161)=A(   87)
 C(  162)=A(   88)
 C(  163)=A(   89)
 Y(   19)=B(   19)/C(  158)
 C(  164)=A(   90)/C(  158)
 C(  165)=A(   91)/C(  158)
 C(  166)=A(   92)/C(  158)
 C(  167)=A(   93)/C(  158)
 C(  168)=A(   94)/C(  158)
 C(  169)=A(   95)/C(  158)
 C(  170)=A(   96)-C(  159)*C(  164)
 C(  171)=A(   97)-C(  160)*C(  164)
 C(  172)=A(   98)-C(  161)*C(  164)
 C(  173)=A(   99)-C(  162)*C(  164)
 C(  174)=A(  100)
 C(  175)=-C(  163)*C(  164)
 Y(   20)=(-C(  159)*Y(   19)
*+B(   20))/C(  170)
 C(  176)=(A(  101)-C(  159)*C(  165)
*)/C(  170)
 C(  177)=(A(  102)-C(  159)*C(  166)
*)/C(  170)
 C(  178)=(A(  103)-C(  159)*C(  167)
*)/C(  170)
 C(  179)=(A(  104)-C(  159)*C(  168)
*)/C(  170)
 C(  180)=(A(  105)-C(  159)*C(  169)
*)/C(  170)
 C(  181)=A(  106)-C(  160)*C(  165)-C(  171)*C(  176)
```

```
 C(  182)=A(  107)-C(  161)*C(  165)-C(  172)*C(  176)
 C(  183)=A(  108)-C(  162)*C(  165)-C(  173)*C(  176)
 C(  184)=-C(  174)*C(  176)
 C(  185)=-C(  163)*C(  165)-C(  175)*C(  176)
 Y(   21)=(-C(  160)*Y(   19)-C(  171)*Y(   20)
*+B(   21))/C(  181)
 C(  186)=(A(  109)-C(  160)*C(  166)-C(  171)*C(  177)
*)/C(  181)
 C(  187)=A(  110)/C(  181)
 C(  188)=(A(  111)-C(  160)*C(  167)-C(  171)*C(  178)
*)/C(  181)
 C(  189)=(-C(  160)*C(  168)-C(  171)*C(  179)
*)/C(  181)
 C(  190)=(-C(  160)*C(  169)-C(  171)*C(  180)
*)/C(  181)
 C(  191)=A(  112)-C(  161)*C(  166)-C(  172)*C(  177)
*-C(  182)*C(  186)
 C(  192)=A(  113)
 C(  193)=A(  114)-C(  162)*C(  166)-C(  173)*C(  177)
*-C(  183)*C(  186)
 C(  194)=-C(  174)*C(  177)-C(  184)*C(  186)
 C(  195)=-C(  163)*C(  166)-C(  175)*C(  177)-C(  185)*C(  186)
 Y(   22)=(-C(  161)*Y(   19)-C(  172)*Y(   20)-C(  182)*Y(   21)
*+B(   22))/C(  191)
 C(  196)=(A(  115)-C(  182)*C(  187)
*)/C(  191)
 C(  197)=(A(  116)-C(  161)*C(  167)-C(  172)*C(  178)
*-C(  182)*C(  188)
*)/C(  191)
 C(  198)=(-C(  161)*C(  168)-C(  172)*C(  179)-C(  182)*C(  189)
*)/C(  191)
 C(  199)=(-C(  161)*C(  169)-C(  172)*C(  180)-C(  182)*C(  190)
*)/C(  191)
 C(  200)=A(  117)-C(  192)*C(  196)
 C(  201)=A(  118)
 C(  202)=A(  119)-C(  183)*C(  187)-C(  193)*C(  196)
 C(  203)=-C(  184)*C(  187)-C(  194)*C(  196)
 C(  204)=-C(  185)*C(  187)-C(  195)*C(  196)
 Y(   23)=(-C(  192)*Y(   22)
*+B(   23))/C(  200)
 C(  205)=A(  120)/C(  200)
 C(  206)=(A(  121)-C(  192)*C(  197)
*)/C(  200)
 C(  207)=(-C(  192)*C(  198)
*)/C(  200)
 C(  208)=(-C(  192)*C(  199)
*)/C(  200)
 C(  209)=A(  122)-C(  201)*C(  205)
 C(  210)=A(  123)
 C(  211)=A(  124)-C(  202)*C(  205)
 C(  212)=-C(  203)*C(  205)
 C(  213)=-C(  204)*C(  205)
 Y(   24)=(-C(  201)*Y(   23)
*+B(   24))/C(  209)
 C(  214)=A(  125)/C(  209)
 C(  215)=(A(  126)-C(  201)*C(  206)
*)/C(  209)
 C(  216)=(-C(  201)*C(  207)
*)/C(  209)
 C(  217)=(-C(  201)*C(  208)
*)/C(  209)
 C(  218)=A(  127)-C(  210)*C(  214)
```

```
      C(  219)=A(  128)
      C(  220)=A(  129)-C(  211)*C(  214)
      C(  221)=-C(  212)*C(  214)
      C(  222)=-C(  213)*C(  214)
      Y(   25)=(-C(  210)*Y(   24)
     *+B(   25))/C(  218)
      C(  223)=A(  130)/C(  218)
      C(  224)=(A(  131)-C(  210)*C(  215)
     *)/C(  218)
      C(  225)=(-C(  210)*C(  216)
     *)/C(  218)
      C(  226)=(-C(  210)*C(  217)
     *)/C(  218)
      C(  227)=A(  132)-C(  219)*C(  223)
      C(  228)=A(  133)
      C(  229)=A(  134)-C(  220)*C(  223)
      C(  230)=-C(  221)*C(  223)
      C(  231)=-C(  222)*C(  223)
      Y(   26)=(-C(  219)*Y(   25)
     *+B(   26))/C(  227)
      C(  232)=A(  135)/C(  227)
      C(  233)=(A(  136)-C(  219)*C(  224)
     *)/C(  227)
      C(  234)=(-C(  219)*C(  225)
     *)/C(  227)
      C(  235)=(-C(  219)*C(  226)
     *)/C(  227)
      C(  236)=A(  137)-C(  228)*C(  232)
      C(  237)=A(  138)
      C(  238)=A(  139)-C(  229)*C(  232)
      C(  239)=-C(  230)*C(  232)
      C(  240)=-C(  231)*C(  232)
      Y(   27)=(-C(  228)*Y(   26)
     *+B(   27))/C(  236)
      C(  241)=A(  140)/C(  236)
      C(  242)=(A(  141)-C(  228)*C(  233)
     *)/C(  236)
      C(  243)=(-C(  228)*C(  234)
     *)/C(  236)
      C(  244)=(-C(  228)*C(  235)
     *)/C(  236)
      C(  245)=A(  142)-C(  237)*C(  241)
      C(  246)=A(  143)-C(  238)*C(  241)
      C(  247)=-C(  239)*C(  241)
      C(  248)=-C(  240)*C(  241)
      Y(   28)=(-C(  237)*Y(   27)
     *+B(   28))/C(  245)
      C(  249)=(A(  144)-C(  237)*C(  242)
     *)/C(  245)
      C(  250)=(-C(  237)*C(  243)
     *)/C(  245)
      C(  251)=(-C(  237)*C(  244)
     *)/C(  245)
      C(  252)=A(  145)-C(  162)*C(  167)-C(  173)*C(  178)
     *-C(  183)*C(  188)-C(  193)*C(  197)-C(  202)*C(  206)
     *-C(  211)*C(  215)-C(  220)*C(  224)-C(  229)*C(  233)
     *-C(  238)*C(  242)-C(  246)*C(  249)
      C(  253)=-C(  174)*C(  178)-C(  184)*C(  188)-C(  194)*C(  197)
     *-C(  203)*C(  206)-C(  212)*C(  215)-C(  221)*C(  224)
     *-C(  230)*C(  233)-C(  239)*C(  242)-C(  247)*C(  249)
      C(  254)=-C(  163)*C(  167)-C(  175)*C(  178)-C(  185)*C(  188)
     *-C(  195)*C(  197)-C(  204)*C(  206)-C(  213)*C(  215)
```

```
*-C(  222)*C(  224)-C(  231)*C(  233)-C(  240)*C(  242)
*-C(  248)*C(  249)
 Y(   29)=(-C(  162)*Y(   19)-C(  173)*Y(   20)-C(  183)*Y(   21)
*-C(  193)*Y(   22)-C(  202)*Y(   23)-C(  211)*Y(   24)
*-C(  220)*Y(   25)-C(  229)*Y(   26)-C(  238)*Y(   27)
*-C(  246)*Y(   28)
*+B(   29))/C(  252)
 C(  255)=(-C(  162)*C(  168)-C(  173)*C(  179)-C(  183)*C(  189)
*-C(  193)*C(  198)-C(  202)*C(  207)-C(  211)*C(  216)
*-C(  220)*C(  225)-C(  229)*C(  234)-C(  238)*C(  243)
*-C(  246)*C(  250)
*)/C(  252)
 C(  256)=(-C(  162)*C(  169)-C(  173)*C(  180)-C(  183)*C(  190)
*-C(  193)*C(  199)-C(  202)*C(  208)-C(  211)*C(  217)
*-C(  220)*C(  226)-C(  229)*C(  235)-C(  238)*C(  244)
*-C(  246)*C(  251)
*)/C(  252)
 C(  257)=A(  146)-C(  145)*C(  147)-C(  149)*C(  150)
*-C(  152)*C(  155)-C(  174)*C(  179)-C(  184)*C(  189)
*-C(  194)*C(  198)-C(  203)*C(  207)-C(  212)*C(  216)
*-C(  221)*C(  225)-C(  230)*C(  234)-C(  239)*C(  243)
*-C(  247)*C(  250)-C(  253)*C(  255)
 C(  258)=A(  147)-C(  163)*C(  168)-C(  175)*C(  179)
*-C(  185)*C(  189)-C(  195)*C(  198)-C(  204)*C(  207)
*-C(  213)*C(  216)-C(  222)*C(  225)-C(  231)*C(  234)
*-C(  240)*C(  243)-C(  248)*C(  250)-C(  254)*C(  255)
 C(  259)=-C(  153)*C(  155)
 C(  260)=-C(  154)*C(  155)
 Y(   30)=(-C(  145)*Y(   16)-C(  149)*Y(   17)-C(  152)*Y(   18)
*-C(  174)*Y(   20)-C(  184)*Y(   21)-C(  194)*Y(   22)
*-C(  203)*Y(   23)-C(  212)*Y(   24)-C(  221)*Y(   25)
*-C(  230)*Y(   26)-C(  239)*Y(   27)-C(  247)*Y(   28)
*-C(  253)*Y(   29)
*+B(   30))/C(  257)
 C(  261)=(A(  148)-C(  174)*C(  180)-C(  184)*C(  190)
*-C(  194)*C(  199)-C(  203)*C(  208)-C(  212)*C(  217)
*-C(  221)*C(  226)-C(  230)*C(  235)-C(  239)*C(  244)
*-C(  247)*C(  251)-C(  253)*C(  256)
*)/C(  257)
 C(  262)=(-C(  152)*C(  156)
*)/C(  257)
 C(  263)=(-C(  152)*C(  157)
*)/C(  257)
 C(  264)=A(  149)-C(  163)*C(  169)-C(  175)*C(  180)
*-C(  185)*C(  190)-C(  195)*C(  199)-C(  204)*C(  208)
*-C(  213)*C(  217)-C(  222)*C(  226)-C(  231)*C(  235)
*-C(  240)*C(  244)-C(  248)*C(  251)-C(  254)*C(  256)
*-C(  258)*C(  261)
 C(  265)=-C(  259)*C(  261)
 C(  266)=-C(  260)*C(  261)
 Y(   31)=(-C(  163)*Y(   19)-C(  175)*Y(   20)-C(  185)*Y(   21)
*-C(  195)*Y(   22)-C(  204)*Y(   23)-C(  213)*Y(   24)
*-C(  222)*Y(   25)-C(  231)*Y(   26)-C(  240)*Y(   27)
*-C(  248)*Y(   28)-C(  254)*Y(   29)-C(  258)*Y(   30)
*+B(   31))/C(  264)
 C(  267)=(-C(  258)*C(  262)
*)/C(  264)
 C(  268)=(-C(  258)*C(  263)
*)/C(  264)
 C(  269)=A(  150)
 C(  270)=A(  151)
 C(  271)=A(  152)
```

```
      C(  272)=A(  153)
      C(  273)=A(  154)
      C(  274)=A(  155)
      Y(   32)=B(   32)/C(  269)
      C(  275)=A(  156)/C(  269)
      C(  276)=A(  157)/C(  269)
      C(  277)=A(  158)/C(  269)
      C(  278)=A(  159)/C(  269)
      C(  279)=A(  160)/C(  269)
      C(  280)=A(  161)/C(  269)
      C(  281)=A(  162)-C(  270)*C(  275)
      C(  282)=A(  163)-C(  271)*C(  275)
      C(  283)=A(  164)-C(  272)*C(  275)
      C(  284)=A(  165)-C(  273)*C(  275)
      C(  285)=A(  166)
      C(  286)=-C(  274)*C(  275)
      Y(   33)=(-C(  270)*Y(   32)
     *+B(   33))/C(  281)
      C(  287)=(A(  167)-C(  270)*C(  276)
     *)/C(  281)
      C(  288)=(A(  168)-C(  270)*C(  277)
     *)/C(  281)
      C(  289)=(A(  169)-C(  270)*C(  278)
     *)/C(  281)
      C(  290)=(A(  170)-C(  270)*C(  279)
     *)/C(  281)
      C(  291)=(A(  171)-C(  270)*C(  280)
     *)/C(  281)
      C(  292)=A(  172)-C(  271)*C(  276)-C(  282)*C(  287)
      C(  293)=A(  173)-C(  272)*C(  276)-C(  283)*C(  287)
      C(  294)=A(  174)-C(  273)*C(  276)-C(  284)*C(  287)
      C(  295)=-C(  285)*C(  287)
      C(  296)=-C(  274)*C(  276)-C(  286)*C(  287)
      Y(   34)=(-C(  271)*Y(   32)-C(  282)*Y(   33)
     *+B(   34))/C(  292)
      C(  297)=(A(  175)-C(  271)*C(  277)-C(  282)*C(  288)
     *)/C(  292)
      C(  298)=A(  176)/C(  292)
      C(  299)=(A(  177)-C(  271)*C(  278)-C(  282)*C(  289)
     *)/C(  292)
      C(  300)=(-C(  271)*C(  279)-C(  282)*C(  290)
     *)/C(  292)
      C(  301)=(-C(  271)*C(  280)-C(  282)*C(  291)
     *)/C(  292)
      C(  302)=A(  178)-C(  272)*C(  277)-C(  283)*C(  288)
     *-C(  293)*C(  297)
      C(  303)=A(  179)
      C(  304)=A(  180)-C(  273)*C(  277)-C(  284)*C(  288)
     *-C(  294)*C(  297)
      C(  305)=-C(  285)*C(  288)-C(  295)*C(  297)
      C(  306)=-C(  274)*C(  277)-C(  286)*C(  288)-C(  296)*C(  297)
      Y(   35)=(-C(  272)*Y(   32)-C(  283)*Y(   33)-C(  293)*Y(   34)
     *+B(   35))/C(  302)
      C(  307)=(A(  181)-C(  293)*C(  298)
     *)/C(  302)
      C(  308)=(A(  182)-C(  272)*C(  278)-C(  283)*C(  289)
     *-C(  293)*C(  299)
     *)/C(  302)
      C(  309)=(-C(  272)*C(  279)-C(  283)*C(  290)-C(  293)*C(  300)
     *)/C(  302)
      C(  310)=(-C(  272)*C(  280)-C(  283)*C(  291)-C(  293)*C(  301)
     *)/C(  302)
```

```
      C(  311)=A(  183)-C(  303)*C(  307)
      C(  312)=A(  184)
      C(  313)=A(  185)-C(  294)*C(  298)-C(  304)*C(  307)
      C(  314)=-C(  295)*C(  298)-C(  305)*C(  307)
      C(  315)=-C(  296)*C(  298)-C(  306)*C(  307)
      Y(   36)=(-C(  303)*Y(   35)
     *+B(   36))/C(  311)
      C(  316)=A(  186)/C(  311)
      C(  317)=(A(  187)-C(  303)*C(  308)
     *)/C(  311)
      C(  318)=(-C(  303)*C(  309)
     *)/C(  311)
      C(  319)=(-C(  303)*C(  310)
     *)/C(  311)
      C(  320)=A(  188)-C(  312)*C(  316)
      C(  321)=A(  189)
      C(  322)=A(  190)-C(  313)*C(  316)
      C(  323)=-C(  314)*C(  316)
      C(  324)=-C(  315)*C(  316)
      Y(   37)=(-C(  312)*Y(   36)
     *+B(   37))/C(  320)
      C(  325)=A(  191)/C(  320)
      C(  326)=(A(  192)-C(  312)*C(  317)
     *)/C(  320)
      C(  327)=(-C(  312)*C(  318)
     *)/C(  320)
      C(  328)=(-C(  312)*C(  319)
     *)/C(  320)
      C(  329)=A(  193)-C(  321)*C(  325)
      C(  330)=A(  194)
      C(  331)=A(  195)-C(  322)*C(  325)
      C(  332)=-C(  323)*C(  325)
      C(  333)=-C(  324)*C(  325)
      Y(   38)=(-C(  321)*Y(   37)
     *+B(   38))/C(  329)
      C(  334)=A(  196)/C(  329)
      C(  335)=(A(  197)-C(  321)*C(  326)
     *)/C(  329)
      C(  336)=(-C(  321)*C(  327)
     *)/C(  329)
      C(  337)=(-C(  321)*C(  328)
     *)/C(  329)
      C(  338)=A(  198)-C(  330)*C(  334)
      C(  339)=A(  199)
      C(  340)=A(  200)-C(  331)*C(  334)
      C(  341)=-C(  332)*C(  334)
      C(  342)=-C(  333)*C(  334)
      Y(   39)=(-C(  330)*Y(   38)
     *+B(   39))/C(  338)
      C(  343)=A(  201)/C(  338)
      C(  344)=(A(  202)-C(  330)*C(  335)
     *)/C(  338)
      C(  345)=(-C(  330)*C(  336)
     *)/C(  338)
      C(  346)=(-C(  330)*C(  337)
     *)/C(  338)
      C(  347)=A(  203)-C(  339)*C(  343)
      C(  348)=A(  204)
      C(  349)=A(  205)-C(  340)*C(  343)
      C(  350)=-C(  341)*C(  343)
      C(  351)=-C(  342)*C(  343)
      Y(   40)=(-C(  339)*Y(   39)
```

```
     *+B(   40))/C(  347)
      C(  352)=A(  206)/C(  347)
      C(  353)=(A(  207)-C(  339)*C(  344)
     *)/C(  347)
      C(  354)=(-C(  339)*C(  345)
     *)/C(  347)
      C(  355)=(-C(  339)*C(  346)
     *)/C(  347)
      C(  356)=A(  208)-C(  348)*C(  352)
      C(  357)=A(  209)-C(  349)*C(  352)
      C(  358)=-C(  350)*C(  352)
      C(  359)=-C(  351)*C(  352)
      Y(   41)=(-C(  348)*Y(   40)
     *+B(   41))/C(  356)
      C(  360)=(A(  210)-C(  348)*C(  353)
     *)/C(  356)
      C(  361)=(-C(  348)*C(  354)
     *)/C(  356)
      C(  362)=(-C(  348)*C(  355)
     *)/C(  356)
      C(  363)=A(  211)-C(  273)*C(  278)-C(  284)*C(  289)
     *-C(  294)*C(  299)-C(  304)*C(  308)-C(  313)*C(  317)
     *-C(  322)*C(  326)-C(  331)*C(  335)-C(  340)*C(  344)
     *-C(  349)*C(  353)-C(  357)*C(  360)
      C(  364)=-C(  285)*C(  289)-C(  295)*C(  299)-C(  305)*C(  308)
     *-C(  314)*C(  317)-C(  323)*C(  326)-C(  332)*C(  335)
     *-C(  341)*C(  344)-C(  350)*C(  353)-C(  358)*C(  360)
      C(  365)=-C(  274)*C(  278)-C(  286)*C(  289)-C(  296)*C(  299)
     *-C(  306)*C(  308)-C(  315)*C(  317)-C(  324)*C(  326)
     *-C(  333)*C(  335)-C(  342)*C(  344)-C(  351)*C(  353)
     *-C(  359)*C(  360)
      Y(   42)=(-C(  273)*Y(   32)-C(  284)*Y(   33)-C(  294)*Y(   34)
     *-C(  304)*Y(   35)-C(  313)*Y(   36)-C(  322)*Y(   37)
     *-C(  331)*Y(   38)-C(  340)*Y(   39)-C(  349)*Y(   40)
     *-C(  357)*Y(   41)
     *+B(   42))/C(  363)
      C(  366)=(-C(  273)*C(  279)-C(  284)*C(  290)-C(  294)*C(  300)
     *-C(  304)*C(  309)-C(  313)*C(  318)-C(  322)*C(  327)
     *-C(  331)*C(  336)-C(  340)*C(  345)-C(  349)*C(  354)
     *-C(  357)*C(  361)
     *)/C(  363)
      C(  367)=(-C(  273)*C(  280)-C(  284)*C(  291)-C(  294)*C(  301)
     *-C(  304)*C(  310)-C(  313)*C(  319)-C(  322)*C(  328)
     *-C(  331)*C(  337)-C(  340)*C(  346)-C(  349)*C(  355)
     *-C(  357)*C(  362)
     *)/C(  363)
      C(  368)=A(  212)-C(    5)*C(   13)-C(   21)*C(   28)
     *-C(   63)*C(   69)-C(   76)*C(   82)-C(   94)*C(   99)
     *-C(  106)*C(  112)-C(  119)*C(  124)-C(  135)*C(  139)
     *-C(  153)*C(  156)-C(  259)*C(  262)-C(  265)*C(  267)
     *-C(  285)*C(  290)-C(  295)*C(  300)-C(  305)*C(  309)
     *-C(  314)*C(  318)-C(  323)*C(  327)-C(  332)*C(  336)
     *-C(  341)*C(  345)-C(  350)*C(  354)-C(  358)*C(  361)
     *-C(  364)*C(  366)
      C(  369)=A(  213)-C(   77)*C(   82)-C(   95)*C(   99)
     *-C(  107)*C(  112)-C(  120)*C(  124)-C(  136)*C(  139)
     *-C(  154)*C(  156)-C(  260)*C(  262)-C(  266)*C(  267)
     *-C(  274)*C(  279)-C(  286)*C(  290)-C(  296)*C(  300)
     *-C(  306)*C(  309)-C(  315)*C(  318)-C(  324)*C(  327)
     *-C(  333)*C(  336)-C(  342)*C(  345)-C(  351)*C(  354)
     *-C(  359)*C(  361)-C(  365)*C(  366)
      C(  370)=-C(    6)*C(   13)-C(   22)*C(   28)-C(   64)*C(   69)
```

```
*-C(   78)*C(   82)-C(   96)*C(   99)-C(  108)*C(  112)
*-C(  121)*C(  124)-C(  137)*C(  139)
 C(  371)=-C(  109)*C(  112)-C(  122)*C(  124)-C(  138)*C(  139)
 Y(   43)=(-C(    5)*Y(    1)-C(   21)*Y(    2)-C(   63)*Y(    8)
*-C(   76)*Y(    9)-C(   94)*Y(   11)-C(  106)*Y(   12)
*-C(  119)*Y(   13)-C(  135)*Y(   15)-C(  153)*Y(   18)
*-C(  259)*Y(   30)-C(  265)*Y(   31)-C(  285)*Y(   33)
*-C(  295)*Y(   34)-C(  305)*Y(   35)-C(  314)*Y(   36)
*-C(  323)*Y(   37)-C(  332)*Y(   38)-C(  341)*Y(   39)
*-C(  350)*Y(   40)-C(  358)*Y(   41)-C(  364)*Y(   42)
*+B(   43))/C(  368)
 C(  372)=(A(  214)-C(    5)*C(   14)-C(   21)*C(   29)
*-C(   63)*C(   70)-C(   76)*C(   83)-C(   94)*C(  100)
*-C(  106)*C(  113)-C(  119)*C(  125)-C(  135)*C(  140)
*-C(  153)*C(  157)-C(  259)*C(  263)-C(  265)*C(  268)
*-C(  285)*C(  291)-C(  295)*C(  301)-C(  305)*C(  310)
*-C(  314)*C(  319)-C(  323)*C(  328)-C(  332)*C(  337)
*-C(  341)*C(  346)-C(  350)*C(  355)-C(  358)*C(  362)
*-C(  364)*C(  367)
*)/C(  368)
 C(  373)=(-C(    5)*C(   15)-C(   21)*C(   30)-C(   63)*C(   71)
*-C(   76)*C(   84)-C(   94)*C(  101)-C(  106)*C(  114)
*-C(  119)*C(  126)-C(  135)*C(  141)
*)/C(  368)
 C(  374)=(-C(  106)*C(  115)-C(  119)*C(  127)-C(  135)*C(  142)
*)/C(  368)
 C(  375)=A(  215)-C(   77)*C(   83)-C(   95)*C(  100)
*-C(  107)*C(  113)-C(  120)*C(  125)-C(  136)*C(  140)
*-C(  154)*C(  157)-C(  260)*C(  263)-C(  266)*C(  268)
*-C(  274)*C(  280)-C(  286)*C(  291)-C(  296)*C(  301)
*-C(  306)*C(  310)-C(  315)*C(  319)-C(  324)*C(  328)
*-C(  333)*C(  337)-C(  342)*C(  346)-C(  351)*C(  355)
*-C(  359)*C(  362)-C(  365)*C(  367)-C(  369)*C(  372)
 C(  376)=-C(    6)*C(   14)-C(   22)*C(   29)-C(   64)*C(   70)
*-C(   78)*C(   83)-C(   96)*C(  100)-C(  108)*C(  113)
*-C(  121)*C(  125)-C(  137)*C(  140)-C(  370)*C(  372)
 C(  377)=-C(  109)*C(  113)-C(  122)*C(  125)-C(  138)*C(  140)
*-C(  371)*C(  372)
 Y(   44)=(-C(   77)*Y(    9)-C(   95)*Y(   11)-C(  107)*Y(   12)
*-C(  120)*Y(   13)-C(  136)*Y(   15)-C(  154)*Y(   18)
*-C(  260)*Y(   30)-C(  266)*Y(   31)-C(  274)*Y(   32)
*-C(  286)*Y(   33)-C(  296)*Y(   34)-C(  306)*Y(   35)
*-C(  315)*Y(   36)-C(  324)*Y(   37)-C(  333)*Y(   38)
*-C(  342)*Y(   39)-C(  351)*Y(   40)-C(  359)*Y(   41)
*-C(  365)*Y(   42)-C(  369)*Y(   43)
*+B(   44))/C(  375)
 C(  378)=(-C(   77)*C(   84)-C(   95)*C(  101)-C(  107)*C(  114)
*-C(  120)*C(  126)-C(  136)*C(  141)-C(  369)*C(  373)
*)/C(  375)
 C(  379)=(-C(  107)*C(  115)-C(  120)*C(  127)-C(  136)*C(  142)
*-C(  369)*C(  374)
*)/C(  375)
 C(  380)=A(  216)-C(    6)*C(   15)-C(   22)*C(   30)
*-C(   64)*C(   71)-C(   78)*C(   84)-C(   96)*C(  101)
*-C(  108)*C(  114)-C(  121)*C(  126)-C(  137)*C(  141)
*-C(  370)*C(  373)-C(  376)*C(  378)
 C(  381)=A(  217)-C(  109)*C(  114)-C(  122)*C(  126)
*-C(  138)*C(  141)-C(  371)*C(  373)-C(  377)*C(  378)
 C(  382)=A(  218)
 C(  383)=A(  219)
 Y(   45)=(-C(    6)*Y(    1)-C(   22)*Y(    2)-C(   64)*Y(    8)
*-C(   78)*Y(    9)-C(   96)*Y(   11)-C(  108)*Y(   12)
```

```
*-C(  121)*Y(   13)-C(  137)*Y(   15)-C(  370)*Y(   43)
*-C(  376)*Y(   44)
*+B(   45))/C(  380)
 C(  384)=(A(  220)-C(  108)*C(  115)-C(  121)*C(  127)
*-C(  137)*C(  142)-C(  370)*C(  374)-C(  376)*C(  379)
*)/C(  380)
 C(  385)=A(  221)/C(  380)
 C(  386)=A(  222)-C(  109)*C(  115)-C(  122)*C(  127)
*-C(  138)*C(  142)-C(  371)*C(  374)-C(  377)*C(  379)
*-C(  381)*C(  384)
 C(  387)=A(  223)-C(  382)*C(  384)
 C(  388)=A(  224)-C(  383)*C(  384)
 Y(   46)=(-C(  109)*Y(   12)-C(  122)*Y(   13)-C(  138)*Y(   15)
*-C(  371)*Y(   43)-C(  377)*Y(   44)-C(  381)*Y(   45)
*+B(   46))/C(  386)
 C(  389)=A(  225)/C(  386)
 C(  390)=(-C(  381)*C(  385)
*)/C(  386)
 C(  391)=A(  226)-C(  387)*C(  389)
 C(  392)=A(  227)-C(  388)*C(  389)
 C(  393)=A(  228)
 C(  394)=A(  229)
 C(  395)=A(  230)
 Y(   47)=(-C(  382)*Y(   45)-C(  387)*Y(   46)
*+B(   47))/C(  391)
 C(  396)=(A(  231)-C(  382)*C(  385)-C(  387)*C(  390)
*)/C(  391)
 C(  397)=A(  232)/C(  391)
 C(  398)=A(  233)/C(  391)
 C(  399)=A(  234)/C(  391)
 C(  400)=A(  235)-C(  383)*C(  385)-C(  388)*C(  390)
*-C(  392)*C(  396)
 C(  401)=A(  236)-C(  393)*C(  396)
 C(  402)=A(  237)-C(  394)*C(  396)
 C(  403)=A(  238)-C(  395)*C(  396)
 Y(   48)=(-C(  383)*Y(   45)-C(  388)*Y(   46)-C(  392)*Y(   47)
*+B(   48))/C(  400)
 C(  404)=(A(  239)-C(  392)*C(  397)
*)/C(  400)
 C(  405)=(A(  240)-C(  392)*C(  398)
*)/C(  400)
 C(  406)=(A(  241)-C(  392)*C(  399)
*)/C(  400)
 C(  407)=A(  242)-C(  393)*C(  397)-C(  401)*C(  404)
 C(  408)=A(  243)-C(  394)*C(  397)-C(  402)*C(  404)
 C(  409)=A(  244)-C(  395)*C(  397)-C(  403)*C(  404)
 Y(   49)=(-C(  393)*Y(   47)-C(  401)*Y(   48)
*+B(   49))/C(  407)
 C(  410)=(A(  245)-C(  393)*C(  398)-C(  401)*C(  405)
*)/C(  407)
 C(  411)=A(  246)/C(  407)
 C(  412)=(A(  247)-C(  393)*C(  399)-C(  401)*C(  406)
*)/C(  407)
 C(  413)=A(  248)-C(  394)*C(  398)-C(  402)*C(  405)
*-C(  408)*C(  410)
 C(  414)=A(  249)
 C(  415)=A(  250)-C(  395)*C(  398)-C(  403)*C(  405)
*-C(  409)*C(  410)
 Y(   50)=(-C(  394)*Y(   47)-C(  402)*Y(   48)-C(  408)*Y(   49)
*+B(   50))/C(  413)
 C(  416)=(A(  251)-C(  408)*C(  411)
*)/C(  413)
```

```
      C(  417)=(A(  252)-C(  394)*C(  399)-C(  402)*C(  406)
     *-C(  408)*C(  412)
     *)/C(  413)
      C(  418)=A(  253)-C(  414)*C(  416)
      C(  419)=A(  254)
      C(  420)=A(  255)-C(  409)*C(  411)-C(  415)*C(  416)
      Y(   51)=(-C(  414)*Y(   50)
     *+B(   51))/C(  418)
      C(  421)=A(  256)/C(  418)
      C(  422)=(A(  257)-C(  414)*C(  417)
     *)/C(  418)
      C(  423)=A(  258)-C(  419)*C(  421)
      C(  424)=A(  259)
      C(  425)=A(  260)-C(  420)*C(  421)
      Y(   52)=(-C(  419)*Y(   51)
     *+B(   52))/C(  423)
      C(  426)=A(  261)/C(  423)
      C(  427)=(A(  262)-C(  419)*C(  422)
     *)/C(  423)
      C(  428)=A(  263)-C(  424)*C(  426)
      C(  429)=A(  264)
      C(  430)=A(  265)-C(  425)*C(  426)
      Y(   53)=(-C(  424)*Y(   52)
     *+B(   53))/C(  428)
      C(  431)=A(  266)/C(  428)
      C(  432)=(A(  267)-C(  424)*C(  427)
     *)/C(  428)
      C(  433)=A(  268)-C(  429)*C(  431)
      C(  434)=A(  269)
      C(  435)=A(  270)-C(  430)*C(  431)
      Y(   54)=(-C(  429)*Y(   53)
     *+B(   54))/C(  433)
      C(  436)=A(  271)/C(  433)
      C(  437)=(A(  272)-C(  429)*C(  432)
     *)/C(  433)
      C(  438)=A(  273)-C(  434)*C(  436)
      C(  439)=A(  274)
      C(  440)=A(  275)-C(  435)*C(  436)
      Y(   55)=(-C(  434)*Y(   54)
     *+B(   55))/C(  438)
      C(  441)=A(  276)/C(  438)
      C(  442)=(A(  277)-C(  434)*C(  437)
     *)/C(  438)
      C(  443)=A(  278)-C(  439)*C(  441)
      C(  444)=A(  279)-C(  440)*C(  441)
      Y(   56)=(-C(  439)*Y(   55)
     *+B(   56))/C(  443)
      C(  445)=(A(  280)-C(  439)*C(  442)
     *)/C(  443)
      C(  446)=A(  281)-C(  395)*C(  399)-C(  403)*C(  406)
     *-C(  409)*C(  412)-C(  415)*C(  417)-C(  420)*C(  422)
     *-C(  425)*C(  427)-C(  430)*C(  432)-C(  435)*C(  437)
     *-C(  440)*C(  442)-C(  444)*C(  445)
      Y(   57)=(-C(  395)*Y(   47)-C(  403)*Y(   48)-C(  409)*Y(   49)
     *-C(  415)*Y(   50)-C(  420)*Y(   51)-C(  425)*Y(   52)
     *-C(  430)*Y(   53)-C(  435)*Y(   54)-C(  440)*Y(   55)
     *-C(  444)*Y(   56)
     *+B(   57))/C(  446)
      X(   57)=Y(   57)
      X(   56)=Y(   56)-C(  445)*X(   57)
      X(   55)=Y(   55)-C(  442)*X(   57)
     *-C(  441)*X(   56)
```

```
 X(   54)=Y(   54)-C(  437)*X(   57)
*-C(  436)*X(   55)
 X(   53)=Y(   53)-C(  432)*X(   57)
*-C(  431)*X(   54)
 X(   52)=Y(   52)-C(  427)*X(   57)
*-C(  426)*X(   53)
 X(   51)=Y(   51)-C(  422)*X(   57)
*-C(  421)*X(   52)
 X(   50)=Y(   50)-C(  417)*X(   57)
*-C(  416)*X(   51)
 X(   49)=Y(   49)-C(  412)*X(   57)
*-C(  411)*X(   51)-C(  410)*X(   50)
 X(   48)=Y(   48)-C(  406)*X(   57)
*-C(  405)*X(   50)-C(  404)*X(   49)
 X(   47)=Y(   47)-C(  399)*X(   57)
*-C(  398)*X(   50)-C(  397)*X(   49)-C(  396)*X(   48)
 X(   46)=Y(   46)-C(  390)*X(   48)
*-C(  389)*X(   47)
 X(   45)=Y(   45)-C(  385)*X(   48)
*-C(  384)*X(   46)
 X(   44)=Y(   44)-C(  379)*X(   46)
*-C(  378)*X(   45)
 X(   43)=Y(   43)-C(  374)*X(   46)
*-C(  373)*X(   45)-C(  372)*X(   44)
 X(   42)=Y(   42)-C(  367)*X(   44)
*-C(  366)*X(   43)
 X(   41)=Y(   41)-C(  362)*X(   44)
*-C(  361)*X(   43)-C(  360)*X(   42)
 X(   40)=Y(   40)-C(  355)*X(   44)
*-C(  354)*X(   43)-C(  353)*X(   42)-C(  352)*X(   41)
 X(   39)=Y(   39)-C(  346)*X(   44)
*-C(  345)*X(   43)-C(  344)*X(   42)-C(  343)*X(   40)
 X(   38)=Y(   38)-C(  337)*X(   44)
*-C(  336)*X(   43)-C(  335)*X(   42)-C(  334)*X(   39)
 X(   37)=Y(   37)-C(  328)*X(   44)
*-C(  327)*X(   43)-C(  326)*X(   42)-C(  325)*X(   38)
 X(   36)=Y(   36)-C(  319)*X(   44)
*-C(  318)*X(   43)-C(  317)*X(   42)-C(  316)*X(   37)
 X(   35)=Y(   35)-C(  310)*X(   44)
*-C(  309)*X(   43)-C(  308)*X(   42)-C(  307)*X(   36)
 X(   34)=Y(   34)-C(  301)*X(   44)
*-C(  300)*X(   43)-C(  299)*X(   42)-C(  298)*X(   36)
*-C(  297)*X(   35)
 X(   33)=Y(   33)-C(  291)*X(   44)
*-C(  290)*X(   43)-C(  289)*X(   42)-C(  288)*X(   35)
*-C(  287)*X(   34)
 X(   32)=Y(   32)-C(  280)*X(   44)
*-C(  279)*X(   43)-C(  278)*X(   42)-C(  277)*X(   35)
*-C(  276)*X(   34)-C(  275)*X(   33)
 X(   31)=Y(   31)-C(  268)*X(   44)
*-C(  267)*X(   43)
 X(   30)=Y(   30)-C(  263)*X(   44)
*-C(  262)*X(   43)-C(  261)*X(   31)
 X(   29)=Y(   29)-C(  256)*X(   31)
*-C(  255)*X(   30)
 X(   28)=Y(   28)-C(  251)*X(   31)
*-C(  250)*X(   30)-C(  249)*X(   29)
 X(   27)=Y(   27)-C(  244)*X(   31)
*-C(  243)*X(   30)-C(  242)*X(   29)-C(  241)*X(   28)
 X(   26)=Y(   26)-C(  235)*X(   31)
*-C(  234)*X(   30)-C(  233)*X(   29)-C(  232)*X(   27)
 X(   25)=Y(   25)-C(  226)*X(   31)
```

```
     *-C(  225)*X(   30)-C(  224)*X(   29)-C(  223)*X(   26)
      X(   24)=Y(   24)-C(  217)*X(   31)
     *-C(  216)*X(   30)-C(  215)*X(   29)-C(  214)*X(   25)
      X(   23)=Y(   23)-C(  208)*X(   31)
     *-C(  207)*X(   30)-C(  206)*X(   29)-C(  205)*X(   24)
      X(   22)=Y(   22)-C(  199)*X(   31)
     *-C(  198)*X(   30)-C(  197)*X(   29)-C(  196)*X(   23)
      X(   21)=Y(   21)-C(  190)*X(   31)
     *-C(  189)*X(   30)-C(  188)*X(   29)-C(  187)*X(   23)
     *-C(  186)*X(   22)
      X(   20)=Y(   20)-C(  180)*X(   31)
     *-C(  179)*X(   30)-C(  178)*X(   29)-C(  177)*X(   22)
     *-C(  176)*X(   21)
      X(   19)=Y(   19)-C(  169)*X(   31)
     *-C(  168)*X(   30)-C(  167)*X(   29)-C(  166)*X(   22)
     *-C(  165)*X(   21)-C(  164)*X(   20)
      X(   18)=Y(   18)-C(  157)*X(   44)
     *-C(  156)*X(   43)-C(  155)*X(   30)
      X(   17)=Y(   17)-C(  150)*X(   30)
      X(   16)=Y(   16)-C(  147)*X(   30)
     *-C(  146)*X(   18)
      X(   15)=Y(   15)-C(  142)*X(   46)
     *-C(  141)*X(   45)-C(  140)*X(   44)-C(  139)*X(   43)
      X(   14)=Y(   14)-C(  133)*X(   46)
     *-C(  132)*X(   45)-C(  131)*X(   44)-C(  130)*X(   43)
     *-C(  129)*X(   15)
      X(   13)=Y(   13)-C(  127)*X(   46)
     *-C(  126)*X(   45)-C(  125)*X(   44)-C(  124)*X(   43)
     *-C(  123)*X(   15)
      X(   12)=Y(   12)-C(  115)*X(   46)
     *-C(  114)*X(   45)-C(  113)*X(   44)-C(  112)*X(   43)
     *-C(  111)*X(   15)-C(  110)*X(   13)
      X(   11)=Y(   11)-C(  101)*X(   45)
     *-C(  100)*X(   44)-C(   99)*X(   43)-C(   98)*X(   15)
     *-C(   97)*X(   12)
      X(   10)=Y(   10)-C(   88)*X(   12)
     *-C(   87)*X(   11)
      X(    9)=Y(    9)-C(   84)*X(   45)
     *-C(   83)*X(   44)-C(   82)*X(   43)-C(   81)*X(   15)
     *-C(   80)*X(   12)-C(   79)*X(   11)
      X(    8)=Y(    8)-C(   71)*X(   45)
     *-C(   70)*X(   44)-C(   69)*X(   43)-C(   68)*X(   15)
     *-C(   67)*X(   12)-C(   66)*X(   11)-C(   65)*X(    9)
      X(    7)=Y(    7)-C(   57)*X(   46)
     *-C(   56)*X(   12)-C(   55)*X(   11)
      X(    6)=Y(    6)-C(   52)*X(   46)
     *-C(   51)*X(   12)-C(   50)*X(   11)-C(   49)*X(    7)
      X(    5)=Y(    5)-C(   45)*X(   46)
     *-C(   44)*X(   12)-C(   43)*X(   11)-C(   42)*X(    6)
      X(    4)=Y(    4)-C(   38)*X(   14)
      X(    3)=Y(    3)-C(   35)*X(   14)
     *-C(   34)*X(    4)
      X(    2)=Y(    2)-C(   30)*X(   45)
     *-C(   29)*X(   44)-C(   28)*X(   43)-C(   27)*X(   15)
     *-C(   26)*X(   12)-C(   25)*X(   11)-C(   24)*X(    9)
     *-C(   23)*X(    8)
      X(    1)=Y(    1)-C(   15)*X(   45)
     *-C(   14)*X(   44)-C(   13)*X(   43)-C(   12)*X(   15)
     *-C(   11)*X(   12)-C(   10)*X(   11)-C(    9)*X(    9)
     *-C(    8)*X(    8)-C(    7)*X(    2)
      WRITE(6,40)
   40 FORMAT(5X,'THE SOLUTIONS OF EQUATIONS   AX = B  ',////)
```

```
      DO 30 I=1,    57
      WRITE(6,20) I,X(I)
   20 FORMAT(10X,'X(',I5,')=',1PE14.7)
   30 CONTINUE
      GO TO 5
   99 END
```

SAMPLE OUTPUT

FROM PROGRAM RESULT

THE SOLUTIONS OF EQUATIONS AX = B

X(1) = 1.0000000+00
X(2) = 3.0000000+00
X(3) = 5.0000000+00
X(4) = -7.0000000+00
X(5) = 9.0000000+00
X(6) = 2.0000000+00
X(7) = 4.0000000+00
X(8) = 6.0000000+00
X(9) = -8.0000000+00
X(10) = 1.0000000+01
X(11) = 5.0000000+00
X(12) = 1.3000000+01
X(13) = 2.1000000+01
X(14) = 2.9000000+01
X(15) = -7.0000000+00
X(16) = 1.1000000+01
X(17) = 1.5000000+01
X(18) = 1.9000000+01
X(19) = 2.5000000+01
X(20) = 8.0000000+00
X(21) = 2.9000000+01
X(22) = 1.3000000+01
X(23) = -1.0000000+01
X(24) = 4.7000000+01
X(25) = 1.3000000+01
X(26) = 8.0000000+00
X(27) = -1.1000000+01
X(28) = 5.0000000+00
X(29) = -7.0000000+00
X(30) = 1.3000000+01
X(31) = 8.0000000+00
X(32) = -1.7000000+01
X(33) = 2.8000000+01
X(34) = 1.6000000+01
X(35) = 9.0000000+00
X(36) = 1.9000000+01
X(37) = -5.4000000+01
X(38) = -4.0000000+00
X(39) = 3.4000000+01
X(40) = -1.3000000+01
X(41) = 4.6000000+01
X(42) = 8.0000000+00
X(43) = 1.7000000+01
X(44) = -1.9000000+01
X(45) = 1.0000000+01
X(46) = 5.3000000+01
X(47) = -3.8000000+01
X(48) = 2.3000000+01
X(49) = -6.0000000+00
X(50) = 1.4000000+01
X(51) = -1.7000000+01
X(52) = 3.0000000+00
X(53) = -3.0000000+00
X(54) = -1.6000000+01
X(55) = 9.0000000+00
X(56) = 7.0000000+00
X(57) = 6.0000000+00

PROGRAM 2

SOURC1

```
0101:        DIMENSION ICARD(12)
0102:        DIMENSION ISEQ(1000),KSEQ(1000)
0103:        DATA LA,LP,LR/'+A(',')*X(',')'/
0104:        J=0
0105:        READ(5,1) N
0106:      1 FORMAT(I5)
0107:        DO 70 I=1,N
0110:        IC=0
0111:        READ(5,10,END=99) NUM,(ISEQ(K),K=1,NUM)
0112:     10 FORMAT(16I5)
0113:        DO 30 L=1,NUM
0114:        J=J+1
0115:        IF(ISEQ(L).EQ.I) GO TO 20
0116:        IC=IC+1
0117:        KSEQ(IC)=ISEQ(L)
0120:        ISEQ(IC)=J
0121:        GO TO 30
0122:     20 JD=J
0123:     30 CONTINUE
0124:        IF(IC.LE.0) GO TO 60
0125:        IP=IC-2
0126:        IF(IP.LE.0) GO TO 40
0127:        WRITE(2,100) I,I,(LA,ISEQ(K),LP,KSEQ(K),LR,K=1,2)
0130:    100 FORMAT(T7,'X(',I5,')=-(-B(',I5,')',2(A3,I5,A4,I5,A1))
0131:        WRITE(2,120) (LA,ISEQ(K),LP,KSEQ(K),LR,K=3,IC)
0132:    120 FORMAT((T6,'*',3(A3,I5,A4,I5,A1)))
0133:        GO TO 50
0134:     40 WRITE(2,100) I,I,(LA,ISEQ(K),LP,KSEQ(K),LR,K=1,IC)
0135:     50 WRITE(2,110) JD
0136:    110 FORMAT(T6,'*',')/A(',I5,')')
0137:        GO TO 70
0140:     60 WRITE(2,130) I,I,JD
0141:    130 FORMAT(T7,'X(',I5,')=B(',I5,')/A(',I5,')')
0142:     70 CONTINUE
0143:     99 ENDFILE2
0144:        REWIND2
0145:        WRITE (3,200) J,N,N,N
0146:    200 FORMAT(T7,'DIMENSION A(',I5,'),B(',I5,'),X(',I5,'),Y(',I5,')')
0147:        WRITE(3,410) J
0150:    410 FORMAT(T7,'READ(5,20) (A(K),K=1,',I5,')')
0151:        WRITE(3,140)
0152:    140 FORMAT(T4,'20 FORMAT(5E14.7)')
0153:        WRITE(3,420) N
0154:    420 FORMAT(T7,'READ(5,20,END=99) (B(K),K=1,',I5,')')
0155:        WRITE (3,440) J
0156:    440 FORMAT(T7,'WRITE(6,20) (A(K),K=1,',I5,')')
0157:        WRITE(3,450) N
0160:    450 FORMAT(T7,'WRITE(6,20) (B(K),K=1,',I5,')')
0161:        WRITE(3,210) N
0162:    210 FORMAT(T7,'DO 40 L=1,',I5)
0163:        WRITE(3,211)
0164:    211 FORMAT(T7,'X(L)=0.')
```

```
0165:         WRITE(3,220)
0166:     220 FORMAT(T4,'40 Y(L)=0.')
0167:         WRITE(3,225)
0170:     225 FORMAT(T7,'ICOUNT=0')
0171:         WRITE(3,230)
0172:     230 FORMAT(T4,'50 ICOUNT=ICOUNT+1')
0173:         WRITE(3,240)
0174:     240 FORMAT(T7,'IF(ICOUNT.GT.25) GO TO 70')
0175:     102 READ(2,101,END=999) (ICARD(K),K=1,12)
0176:     101 FORMAT(12A6)
0177:         WRITE(3,101) (ICARD(K),K=1,12)
0200:         GO TO 102
0201:     999 CONTINUE
0202:         WRITE(3,250) N
0203:     250 FORMAT(T7,'DO 60 L=1,',I5)
0204:         WRITE(3,260)
0205:     260 FORMAT(T7,'TEMP=ABS(Y(L)-X(L))')
0206:         WRITE(3,270)
0207:     270 FORMAT(T7,'IF(TEMP.GT.1E-04) GO TO 80')
0210:         WRITE(3,280)
0211:     280 FORMAT(T4,'60 CONTINUE')
0212:         WRITE(3,290) N
0213:     290 FORMAT(T4,'70 WRITE(6,20) (X(K),K=1,',I5,')')
0214:         WRITE(3,300)
0215:     300 FORMAT(T7,'STOP')
0216:         WRITE(3,310) N
0217:     310 FORMAT(T4,'80 DO 90 L=1,',I5)
0220:         WRITE(3,320)
0221:     320 FORMAT(T7,'Y(L)=X(L)')
0222:         WRITE(3,330)
0223:     330 FORMAT(T4,'90 CONTINUE')
0224:         WRITE(3,340)
0225:     340 FORMAT(T7,'GO TO 50')
0226:         WRITE(3,350)
0227:     350 FORMAT(T4,'99 END')
0230:         ENDFILE 3
0231:         REWIND 3
0232:      11 READ(3,101,END=9999) (ICARD(K),K=1,12)
0233:         WRITE(6,101) (ICARD(I),I=1,12)
0234:         GO TO 11
0235:   9999 STOP
0236:         END
```

OUTPUT OF PROGRAM SOURC1

SYMBOLIC GENERATED PROGRAM

RESUL1

```
101      1*          REAL*4    A(  148),B(   50),X(   50),Y(   50)
103      2*          READ(5,20) (A(K),K=1,  148)
111      3*       20 FORMAT(5E14.7)
112      4*          READ(5,20,END=99) (B(K),K=1,   50)
120      5*          DO 40 L=1,   50
123      6*          X(L)=0.
124      7*       40 Y(L)=0.
126      8*          ICOUNT=0
127      9*       50 ICOUNT=ICOUNT+1
130     10*          IF(ICOUNT.GT.300) GO TO   70
132     11*          X(    1)=-(-B(    1)+A(    2)*X(    2)
132     12*         *)/A(    1)
133     13*          X(    2)=-(-B(    2)+A(    3)*X(    1)+A(    5)*X(    3)
133     14*         *)/A(    4)
134     15*          X(    3)=-(-B(    3)+A(    6)*X(    2)+A(    8)*X(    4)
134     16*         *)/A(    7)
135     17*          X(    4)=-(-B(    4)+A(    9)*X(    3)+A(   11)*X(    5)
135     18*         *)/A(   10)
136     19*          X(    5)=-(-B(    5)+A(   12)*X(    4)+A(   14)*X(    6)
136     20*         *)/A(   13)
137     21*          X(    6)=-(-B(    6)+A(   15)*X(    5)+A(   17)*X(    7)
137     22*         *)/A(   16)
140     23*          X(    7)=-(-B(    7)+A(   18)*X(    6)+A(   20)*X(    8)
140     24*         *)/A(   19)
141     25*          X(    8)=-(-B(    8)+A(   21)*X(    7)+A(   23)*X(    9)
141     26*         *)/A(   22)
142     27*          X(    9)=-(-B(    9)+A(   24)*X(    8)+A(   26)*X(   10)
142     28*         *)/A(   25)
143     29*          X(   10)=-(-B(   10)+A(   27)*X(    9)+A(   29)*X(   11)
143     30*         *)/A(   28)
```

```
144     31*          X(   11)=-(-B(   11)+A(   30)*X(   10)+A(   32)*X(   12)
144     32*         *)/A(   31)
145     33*          X(   12)=-(-B(   12)+A(   33)*X(   11)+A(   35)*X(   13)
145     34*         *)/A(   34)
146     35*          X(   13)=-(-B(   13)+A(   36)*X(   12)+A(   38)*X(   14)
146     36*         *)/A(   37)
147     37*          X(   14)=-(-B(   14)+A(   39)*X(   13)+A(   41)*X(   15)
147     38*         *)/A(   40)
150     39*          X(   15)=-(-B(   15)+A(   42)*X(   14)+A(   44)*X(   16)
150     40*         *)/A(   43)
151     41*          X(   16)=-(-B(   16)+A(   45)*X(   15)+A(   47)*X(   17)
151     42*         *)/A(   46)
152     43*          X(   17)=-(-B(   17)+A(   48)*X(   16)+A(   50)*X(   18)
152     44*         *)/A(   49)
153     45*          X(   18)=-(-B(   18)+A(   51)*X(   17)+A(   53)*X(   19)
153     46*         *)/A(   52)
154     47*          X(   19)=-(-B(   19)+A(   54)*X(   18)+A(   56)*X(   20)
154     48*         *)/A(   55)
155     49*          X(   20)=-(-B(   20)+A(   57)*X(   19)+A(   59)*X(   21)
155     50*         *)/A(   58)
156     51*          X(   21)=-(-B(   21)+A(   60)*X(   20)+A(   62)*X(   22)
156     52*         *)/A(   61)
157     53*          X(   22)=-(-B(   22)+A(   63)*X(   21)+A(   65)*X(   23)
157     54*         *)/A(   64)
160     55*          X(   23)=-(-B(   23)+A(   66)*X(   22)+A(   68)*X(   24)
160     56*         *)/A(   67)
161     57*          X(   24)=-(-B(   24)+A(   69)*X(   23)+A(   71)*X(   25)
161     58*         *)/A(   70)
162     59*          X(   25)=-(-B(   25)+A(   72)*X(   24)+A(   74)*X(   26)
162     60*         *)/A(   73)
163     61*          X(   26)=-(-B(   26)+A(   75)*X(   25)+A(   77)*X(   27)
163     62*         *)/A(   76)
164     63*          X(   27)=-(-B(   27)+A(   78)*X(   26)+A(   80)*X(   28)
164     64*         *)/A(   79)
165     65*          X(   28)=-(-B(   28)+A(   81)*X(   27)+A(   83)*X(   29)
165     66*         *)/A(   82)
166     67*          X(   29)=-(-B(   29)+A(   84)*X(   28)+A(   86)*X(   30)
166     68*         *)/A(   85)
167     69*          X(   30)=-(-B(   30)+A(   87)*X(   29)+A(   89)*X(   31)
167     70*         *)/A(   88)
170     71*          X(   31)=-(-B(   31)+A(   90)*X(   30)+A(   92)*X(   32)
170     72*         *)/A(   91)
171     73*          X(   32)=-(-B(   32)+A(   93)*X(   31)+A(   95)*X(   33)
171     74*         *)/A(   94)
172     75*          X(   33)=-(-B(   33)+A(   96)*X(   32)+A(   98)*X(   34)
172     76*         *)/A(   97)
173     77*          X(   34)=-(-B(   34)+A(   99)*X(   33)+A(  101)*X(   35)
173     78*         *)/A(  100)
174     79*          X(   35)=-(-B(   35)+A(  102)*X(   34)+A(  104)*X(   36)
174     80*         *)/A(  103)
175     81*          X(   36)=-(-B(   36)+A(  105)*X(   35)+A(  107)*X(   37)
175     82*         *)/A(  106)
176     83*          X(   37)=-(-B(   37)+A(  108)*X(   36)+A(  110)*X(   38)
176     84*         *)/A(  109)
177     85*          X(   38)=-(-B(   38)+A(  111)*X(   37)+A(  113)*X(   39)
177     86*         *)/A(  112)
200     87*          X(   39)=-(-B(   39)+A(  114)*X(   38)+A(  116)*X(   40)
200     88*         *)/A(  115)
201     89*          X(   40)=-(-B(   40)+A(  117)*X(   39)+A(  119)*X(   41)
201     90*         *)/A(  118)
202     91*          X(   41)=-(-B(   41)+A(  120)*X(   40)+A(  122)*X(   42)
202     92*         *)/A(  121)
```

```
0203     93*          X(   42)=-(-B(   42)+A(  123)*X(   41)+A(  125)*X(   43)
0203     94*         *)/A(  124)
0204     95*          X(   43)=-(-B(   43)+A(  126)*X(   42)+A(  128)*X(   44)
0204     96*         *)/A(  127)
0205     97*          X(   44)=-(-B(   44)+A(  129)*X(   43)+A(  131)*X(   45)
0205     98*         *)/A(  130)
0206     99*          X(   45)=-(-B(   45)+A(  132)*X(   44)+A(  134)*X(   46)
0206    100*         *)/A(  133)
0207    101*          X(   46)=-(-B(   46)+A(  135)*X(   45)+A(  137)*X(   47)
0207    102*         *)/A(  136)
0210    103*          X(   47)=-(-B(   47)+A(  138)*X(   46)+A(  140)*X(   48)
0210    104*         *)/A(  139)
0211    105*          X(   48)=-(-B(   48)+A(  141)*X(   47)+A(  143)*X(   49)
0211    106*         *)/A(  142)
0212    107*          X(   49)=-(-B(   49)+A(  144)*X(   48)+A(  146)*X(   50)
0212    108*         *)/A(  145)
0213    109*          X(   50)=-(-B(   50)+A(  147)*X(   49)
0213    110*         *)/A(  148)
0214    111*           DO 60 L=1,   50
0217    112*           TEMP=ABS(Y(I)-X(L))
0220    113*           IF(TEMP.GT.1E-04) GO TO 80
0222    114*        60 CONTINUE
0224    115*        70 DO 75 I=1,   50
0227    116*           WRITE(6,85) I,X(I)
0233    117*        85 FORMAT(25X,'X(',I5,')=',E14.7)
0234    118*        75 CONTINUE
0236    119*        25 FORMAT(20X,' THE ITERATIVE METHOD FAILED')
0237    120*           STOP
0240    121*        80 DO 90 L=1,   50
0243    122*           Y(L)=X(L)
0244    123*        90 CONTINUE
0246    124*           GO TO 50
0247    125*        99 END
```

SAMPLE OUTPUT

FROM PROGRAM RESUL1

```
X(    1)=    .1000199+01
X(    2)=    .9996128+00
X(    3)=    .1000559+01
X(    4)=    .9992901+00
X(    5)=    .1000838+01
X(    6)=    .9990578+00
X(    7)=    .1001021+01
X(    8)=    .9989248+00
X(    9)=    .1001106+01
X(   10)=    .9988853+00
X(   11)=    .1001104+01
X(   12)=    .9989233+00
X(   13)=    .1001035+01
X(   14)=    .9990169+00
X(   15)=    .1000923+01
X(   16)=    .9991425+00
X(   17)=    .1000790+01
X(   18)=    .9992772+00
X(   19)=    .1000659+01
X(   20)=    .9993982+00
X(   21)=    .1000554+01
X(   22)=    .9994798+00
X(   23)=    .1000503+01
X(   24)=    .9994916+00
X(   25)=    .1000540+01
X(   26)=    .9993983+00
X(   27)=    .1000699+01
X(   28)=    .9991650+00
X(   29)=    .1001012+01
X(   30)=    .9987705+00
X(   31)=    .1001487+01
X(   32)=    .9982218+00
X(   33)=    .1002097+01
X(   34)=    .9975678+00
X(   35)=    .1002772+01
X(   36)=    .9969001+00
X(   37)=    .1003401+01
X(   38)=    .9963415+00
X(   39)=    .1003857+01
X(   40)=    .9960191+00
X(   41)=    .1004020+01
X(   42)=    .9960333+00
X(   43)=    .1003816+01
X(   44)=    .9964303+00
X(   45)=    .1003232+01
X(   46)=    .9971883+00
X(   47)=    .1002322+01
X(   48)=    .9982217+00
X(   49)=    .1001198+01
X(   50)=    .9994011+00
```