(NASA-TM-X-73074) NASA-AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS SYSTEM (POTFAN) EQUATION SOLVER CODE (SOLN) VERSION 1 (NASA) 154 p HC $6.75 CSCL 09B

N76-32877

Unclas
G3/61 03445

NASA TECHNICAL MEMORANDUM

NASA TM X-73,074

NASA TM X-73,074

NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS SYSTEM (POTFAN) EQUATION SOLVER CODE (SOLN) VERSION 1

J. E. Davis
Computer Sciences Corporation
Mountain View, California 94043

W. S. Bonnett
Acurex Corporation
Mountain View, California 94042

and

R. T. Medan
Ames Research Center
Moffett Field, California 94035

July 1976

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No
NASA TM X-73,074		

4. Title and Subtitle	5. Report Date
NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS SYSTEM (POTFAN) EQUATION SOLVER CODE (SOLN) VERSION 1	6. Performing Organization Code
7. Author(s) J. E. Davis,* W. S. Bonnett,** and R. T. Medan†	8. Performing Organization Report No. A-6338
9. Performing Organization Name and Address *Computer Sciences Corp., Mountain View, CA 94043 **Acurex Corporation, Mountain View, CA 94042 †Ames Research Center, Moffett Field, CA 94035	10 Work Unit No 743-04-21 11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546	13. Type of Report and Period Covered Technical Memorandum 14 Sponsoring Agency Code

15 Supplementary Notes

16. Abstract

This document describes a computer program known as SOLN which has been developed as an independent segment of the NASA-Ames three-dimensional potential flow analysis systems of linear algebraic equations by any of several methods including LU decomposition, Householder's method, a partitioning scheme, and a block successive relaxation method. Due to the independent modular nature of the program, it may be used by itself and not necessarily in conjunction with other segments of the POTFAN system.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Simultaneous equations; Linear equations; Algebraic equations; Decomposition; Least squares; Iteration	Unlimited STAR Category - 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	153	$6.25

*For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS SYSTEM (POTFAN)

EQUATION SOLVER CODE (SOLN) VERSION 1

J. E. Davis
Computer Sciences Corporation

W. S. Bonnett
Acurex Corporation

and

R. T. Medan
Ames Research Center

SUMMARY

This document describes a computer program known as SOLN which has been developed as an independent segment of the NASA-Ames three-dimensional potential flow analysis system (POTFAN), and which is used to solve small to large systems of linear algebraic equations by any of several methods including LU decomposition, Householder's method, a partitioning scheme, and a block successive relaxation method. Due to the independent modular nature of the program, it may be used by itself and not necessarily in conjunction with other segments of the POTFAN system.

1 INTRODUCTION

This document describes version 1 of an equation solver computer code (SOLN) which is a segment of the NASA-Ames three-dimensional potential flow analysis system (POTFAN). This segment of the system solves the set of linear algebraic equations that are generated by the collocation method of satisfying the boundary condition of a specified flow at various locations on components in the flow field under consideration.

The sets of constant coefficients and right hand sides associated with these simultaneous equations are calculated by other programs in the POTFAN system and transmitted to the SOLN program as files through auxiliary storage devices. The SOLN code reads in these files, determines the solutions, and then writes them out as files to be read in by the next program in the POTFAN system. See Fig. 1-1.

The SOLN code provides a variety of solution techniques including LU decomposition, Householder's method, a partitioning scheme, and block successive over-relaxation. These various solution techniques are provided so that each different type of system of equations encountered in potential flow aerodynamics may be handled in the most efficient manner available. The LU decomposition method is the fastest technique available for solving a small, closely coupled, square system of equations that can be placed in core all at once; the Householder procedure is the best method for solving an overdetermined system of equations; the partitioning scheme is best for large closely coupled systems of equations that cannot be placed in core all at one time; and the block successive over-relaxation procedure is the potentially fastest method available for large, diagonally dominant systems of equations such as those generated in solving the problem of an aircraft component in a wind tunnel.

The number of equations that can be simultaneously solved by the code is mainly limited only by the amount of tape, disc, or drum storage available to the user.

The SOLN code is constructed in a modular fashion so that any modifications or improvements to particular portions of the code do not affect the rest of the code.

2 PROBLEM TASK DESCRIPTION

This computer program was developed under Task 3 of NASA-Ames Contract NAS2-7571, and Task 26 of NASA-Ames Contract NAS2-6912. The purpose of these tasks was to develop a computer code which will efficiertly solve the systems of linear algebraic equations arising from attempting to satisfy the boundary conditions of a specified flow on the various components in the flow field by the collocation method. These systems of equations may or may not exceed the size of available computer storage, may be partitioned in an arbitrary manner with each partition representing the influence of one component on itself or another component, and may be overdetermined in some cases. In addition, a block iteration procedure was to be provided where each block represents the aerodynamic influence of a single aircraft component on itself or on another aircraft component. This iteration procedure must allow an add-on solution capability without unnecessarily repeating any calculations (for example, obtaining the solution for a wing alone and then adding in wind tunnel walls to determine the wall correction) and must allow an assemblage of components (for example, wing + body + tail) to be solved iteratively. Therefore, a variety of solution procedures were to be developed to best handle each type of situation likely to be encountered.

Also the program was required to be modular in nature so that it could be used independently and so that any modifications or improvements to the code would not affect the other segments of the POTFAN system.

Furthermore, the code was to be constructed so that it is versatile, yet easy to use and easy to modify.

Finally the program was required to be able to handle large problems consisting of many separate components and/or large influence matrices.

3 METHOD OF SOLUTION

This section describes how the problems posed in the previous section were solved.

3.1 EQUATION SOLVING OPTIONS

The SOLN program was designed to compute the solutions of linear algebraic systems. It can compute these solutions by one of several different procedures depending on the particular circumstances of the problems. In general, the system of equations to be solved may be overspecified (more equations than unknowns) or properly specified. These types of systems produce a rectangular or square matrix, respectively. The SOLN code uses a different solution procedure for each. In addition, different procedures are used depending on the size of the system of equations to be solved. One type of procedure is used for systems of equations that are small enough to contain all necessary data within allowable core storage at one time, while other procedures are used when the system of equations is too large to reside in core all at once. The latter procedures make extensive use of temporary storage devices such as tape, disc, or drum files. A special block iteration procedure is also available which best handles problems where certain blocks of a matrix are only weakly coupled with other blocks. This iteration procedure also operates differently depending on whether the system to be solved is rectangular or square in nature, and whether each block fits in core at one time or not.

A detailed discussion of each of the solution procedures available in the SOLN code is contained in the following subsections. In the following discussion, it will be assumed that capital letters stand for matrices of at least two columns and rows, and that lower case letters stand for column vectors. In that case, the problem to be solved can be stated as: given A and B, find X such that

$$AX = B \quad \text{or} \quad Ax = b. \qquad (3.1\text{-}1)$$

The only assumptions that are made about A are that either A or the transpose of A times A is nonsingular. It is also assumed that A, X and B have real elements. No structure assumptions have been made about A: it is not banded in any systematic manner, it is not symmetric or positive definite except by chance, and it is not even necessarily square, although squareness does cut down on execution time.

There are four main procedures used in this code, the LU Decomposition method (or the Crout-Doolittle method), the Householder method, a matrix partitioning scheme for large matrices that was developed especially for this program, and a block Gauss-Siedel iteration procedure. The references provided in the reference section are very helpful in the matrix theory involved. Forsythe (1967) has a Doolittle method written in Algol 60 and Fortran, and provides a glimpse of error considerations. Nering (1963) gives a complete development of matrix theory in a theoretical manner, while Ralston(1963) provided the basis for the presently included decomposition subroutine. Westlake(1968) is perhaps the most valuable for a user interested in selecting and implementing his own procedure.

3.1.1 The Decomposition Method

This is one of the best all around methods for a general, nonbanded, nonsymmetric real matrix. It is a method of solving a set of equations without calculating either the inverse of the matrix or any sequence of large matrices. The procedure is approximately as fast and just as accurate as any equation solving technique so far found. This procedure also gives a clear-cut singularity indicator before the entire procedure has been executed. The double precision version of this routine is very readily arrived at, if desired, as well.

The method is based on decomposing a given matrix A into the matrix product of a unit lower triangular matrix L and an upper triangular matrix U, so that

$$A = LU. \qquad (3.1.1\text{-}1)$$

Once this has been accomplished, the matrix problem becomes a set of two easily solved equations

$$Ly = b \qquad (3.1.1\text{-}2)$$

and

$$Ux = y. \qquad (3.1.1\text{-}3)$$

The decomposition is almost as easily done as said. To produce the first row of U, the first row of A is taken, since the components of the first row of L (which is, $l_{11} = 1$, $l_{12} = 0$, $l_{13} = 0$, $\ldots l_{1n} = 0$) times the columns of U produce the first row of U identically. And, by the equality statement, the first row of U equals the first row of A. Now, with the first row of U available, the first column of L can be calculated. This proceeds to the nth row of U, alternately calculating rows of U and columns of L with data provided by the previous rows and columns. The equations at the rth step are

$$u_{ri} = a_{ri} - \sum_{k=1}^{r-1} l_{rk} u_{ki} \qquad (3.1.1\text{-}4)$$

where

$$l_{mr} = (a_{mr} - \sum_{k=1}^{r-1} l_{mk} u_{kr})/u_{rr} \qquad (3.1.1\text{-}5)$$

This decomposition procedure breaks down at any point it which the diagonal element u_{rr} vanishes. To prevent :his, each time the procedure starts a new row of U, it searches the column under consideration for the largest element. Having identified the largest element on the liagonal, the row about to be reduced and the row containing :he largest element are interchanged. Thus, if a column is ver searched, and nothing but zeros are found, the rocedure stops, since a singular matrix has been found.

The decomposition of the matrix takes roughly $n^3/3$ ultiplications, contrasted to the usual n^3 operations of auss-Jordan reduction, and is therefore much faster. Since he inverse has not been explicitly determined, that extra torage has not been needed since the intermediate results f the decomposition are stored where the matrix was.

The auxiliary equations are solved very quickly ecause they are triangular. First the vector y is etermined, then the solution vector x. The first component of y is b_1, the first component of b. The second component of y is determined from

$$l_{21} y_1 + y_2 = b_2 \qquad (3.1.1\text{-}6)$$

and so on. Once y has been determined, the solution vector is "unzipped" from U in the same manner, only starting from the bottom.

In summary, the equations to be solved are

$$A = LU \qquad (3.1.1\text{-}7)$$

$$Ly = b \qquad (3.1.1\text{-}8)$$

$$Ux = y \qquad (3.1.1\text{-}9)$$

3.1.2 Householder's Method

This method involves triangularization of the matrix and the use of elementary Hermitean unitary transformations. At present, the operations done to the matrix to triangularize it must also be done to the constant vectors simultaneously. In other words, unlike the decomposition method, there are no provisions for storing the operations done to the matrix so that they may be done to constant vectors independently at a later time. Every constant vector must be present at the time of triangularization, or the original matrix must be reread and retriangularized.

The method involves premultiplying a matrix by a sequence of other matrices, all of which are unitary matrices, so that the condition number of the matrix is unchanged at each step. The condition number is a function of the size of the matrix inverse, and determines how badly roundoff effects confuse the answer. It can be shown that the condition number does not change as the matrix is manipulated and therefore gets no worse.

Basically the Householder method seeks to triangularize the matrix A, by multiplying it by a series of the unitary matrices, called P_i. For the first step we would like

$$PA^{(0)} = A^{(1)} \qquad (3.1.2-1)$$

here the first cclumn of $A^{(1)}$ (call it a) has a nonzero lement in the first row, but zeros in all subsequent rows f this cclumn.

$$Pa = ke_1 \quad \text{where } e_1 = (1,0,0,0...0) \qquad (3.1.2-2)$$

he size of "a" can be determined in the usual method:

$$||a|| = \left[\sum_{k=1}^{n} (a_{k1})^2 \right]^{1/2} = s \qquad (3.1.2-3)$$

$$\Rightarrow ||a||^2 = s^2 \qquad (3.1.2-4)$$

: can be shown that the size of a unitary transformation of vector is the same as the vector itself:

$$||Pa||^2 = \left[\sum_{k=1}^{n} (a_{k1})^2 \right] = s^2 \qquad (3.1.2-5)$$

but

$$||Pa||^2 = ||ke_1||^2 = k^2 \qquad (3.1.2-6)$$

and therefore:

$$k = \pm s \qquad (3.1.2-7)$$

It can be shown that for any vector w such that $||w|| =$ and w is real, then a matrix constructed by

$$U = I - 2ww^T \qquad (3.1.2\text{-}8)$$

is unitary. The choice of w with which to construct a matrix of this form so that it has the properties of P is the key to the procedure.

$$Pa = (I - 2ww^T)a = a - 2w(w^T a) \qquad (3.1.2\text{-}9)$$

but the matrix product $w^T a = K$, is a constant. All elements other than the first of this resultant vector should vanish, since the first column is being triangularized. That implies the following set of equations:

$$a_i - 2Kw_i = 0 \quad i \neq 1 \qquad (3.1.2\text{-}10)$$

$$a_1 - 2Kw_1 = k = \pm s \qquad (3.1.2\text{-}11)$$

or

$$w_i = a/2K \quad i \neq 1 \quad \text{and} \quad w_1 = (a_1 \pm s)/2K \quad (3.1.2\text{-}12)$$

but, $||w|| = 1$, by the requirements for constructing the matrix, and therefore

$$\sum_{k=1}^{n} (w_i)^2 = 1 = \frac{1}{4K^2}\left[(a \pm s)^2 + \sum_{k=2}^{n} (a_i)^2 \right] \qquad (3.1.2\text{-}13)$$

$$= \frac{1}{4K^2}\left[a_1^2 \pm 2a_1 s + s^2 + \sum_{k=2}^{n} (a_i)^2 \right]$$

(3.1.2-14)

$$= \frac{1}{4K^2}\left[\quad ||a||^2 \pm 2a_1 s + s^2 \right] \qquad (3.1.2-15)$$

$$= \frac{1}{4K^2}\left[\quad s^2 \pm 2a_1 s + s^2 \right] \qquad (3.1.2-16)$$

$$= \frac{1}{4K^2}\, 2 \left[s^2 \pm a_1 s \right] \qquad (3.1.2-17)$$

Therefore:

$$2K^2 = s^2 \pm a_1 s \qquad (3.1.2-18)$$

f the vector u is defined as:

$$u = ((a_1 \pm s), a_2, a_3 \ldots a_n) \qquad (3.1.2-19)$$

nd w as $w = u/2K$, then the following relations can be shown o be true:

a) $w^T w = 1$ (3.1.2-20)

b) $P = I - 2ww^T$ is unitary (3.1.2-21)

and

c) $Pa = \pm s e_1$. (3.1.2-22)

At this point, the first column of A has been :iangularized. The other columns of the matrix are then)erated on by the unitary matrix (or its vector form), .though none of them are triangularized if the matrix is .ngular. The next step in the procedure is to develop the :ctor w for producing the same triangularization of the second column, from its diagonal element on down. The third column will be treated from the third row down, and so on. In this way, a series of matrices are generated which premultiply the original matrix. The $(n - 1)^{th}$ matrix produces an upper triangular matrix. If the same transformations have been done to the constant vectors, then all that need be done is a back-substitution to produce the solution.

Another important feature of this procedure is the fact that no division by diagonal elements, and therefore no pivoting of rows, need be done. Also, this procedure can be used to produce "least-squared" solutions if the matrix has more rows than columns, without explicitly multiplying the matrices. In other words, the system

$$(A^T A) x = A^T b \qquad (3.1.2-23)$$

can be solved without actually multiplying the matrices A^T and A together.

3.1.3 The Partitioning Schemes

There are several possible methods for partitioning a very large matrix into submatrices in order to create the matrix inverse. Most of these methods could be termed "block" methods, in that they apply well defined methods for dealing with individual elements of a matrix to submatrices. For instance, a very popular method for solving a predictably sparse, well behaved matrix system is the method of successive over-relaxation. The block analog of this is the method of block successive over-relaxation described in section 3.1.4.

In this section, a partitioning scheme designed to take maximum advantage of available core storage is described. In this manual it is known as the large decomposition solution procedure. The partitions developed in this algorithm are not related to the natural partitions associated with the separate components of the configuration.

Basically, the method starts in the upper left corner of the large, dense matrix and inverts directly as large a block as is possible. From that point on, pieces of the original matrix are processed as the matrix inverse "grows" toward the lower right corner. The number of words in each partition is as large as possible. The sequence of the inversion is shown in Figure 3.1.3-1, and the sequence of

:quations at each step are given by

B is replaced by AB

$$D = D - CB \qquad (3.1.3\text{-}1)$$

$$D = D^{-1} \qquad (3.1.3\text{-}2)$$

$$B = -BD \qquad (3.1.3\text{-}3)$$

$$C = CA \qquad (3.1.3\text{-}4)$$

$$A = A - BC \qquad (3.1.3\text{-}5)$$

$$C = -DC \qquad (3.1.3\text{-}6)$$

Let the matrix be partitioned as is shown in Figure .1.3-2. The inverse of that matrix is assumed to be of the ollowing form

$$\begin{bmatrix} LM \\ NP \end{bmatrix} \qquad (3.1.3\text{-}7)$$

Since the product of the matrix and its inverse is the dentity matrix, the following set of equations can be eveloped

$$P = (D - CA^{-1} B)^{-1} \qquad (3.1.3\text{-}8)$$

$$M = -A^{-1} BP \qquad (3.1.3\text{-}9)$$

$$N = -PCA^{-1} \qquad (3.1.3\text{-}10)$$

$$L = A^{-1} - A^{-1} BN. \qquad (3.1.3\text{-}11)$$

ıe program equations are identical except that they are esigned to take as little extra storage as possible.

3.1.4 Block Successive Over-Relaxation

The block successive over-relaxation procedure provides an alternate method for the solution of large systems of equations. It is particularly advantageous for matrices that are block diagonally dominant. This situation occurs, for example, in the case of a wing in a wind tunnel. The method consists of an iteration sequence which operates on an initial guess to the solution of the equation

$$Ax = b \qquad (3.1.4\text{-}1)$$

where the matrix A consists of several submatrices A . A submatrix represents the influence of an aircraft component on itself or on another aircraft component. The figure below illustrates this matrix structure.

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{Bmatrix} \qquad (3.1.4\text{-}2)$$

The subscript n refers to the number of subvectors x_j , each of which represents the singularity strengths of a single component.

In order to begin the iteration procedure, an initial guess for each of the subvectors x_j must be either given as input data or computed or both. If the initial guess is to be entirely computed by the program, the procedure begins by approximating the solution for the first subvector x_1° as the solution to the system of equations

$$A_{11}\, x_1^o = b_1 \qquad (3.1.4-3)$$

where the superscript refers to the iteration number and the superscript "o" refers to the initial guess. The initial guesses to the succeeding subvectors x_i^o are computed by sequentially solving each of the following systems of equations:

$$A_{22}\, x_2^o = b_2 - A_{21}\, x_1^o \qquad (3.1.4.4)$$

$$\vdots$$

$$A_{nn}\, x_n^o = b_n - A_{n1}\, x_1^o - \ldots - A_{nn-1}\, x_{n-1}^o$$

At any step in the above procedure, the initial guess for the sub-vector under consideration may be entered as input data rather than being computed. Once an initial guess has been provided for each solution sub-vector, the succeeding iteration proceeds by sequentially solving each of the following systems of equations:

$$A_{11}\, x_1^k = b_1 - A_{12}\, x_2^{k-1} - \ldots - A_{1n}\, x_n^{k-1} \qquad (3.1.4-5)$$

$$A_{22}\, x_2^k = b_2 - A_{21}\, x_1^k - A_{23}\, x_3^{k-1} - \ldots - A_{2n}\, x_n^{k-1}$$

$$\vdots$$

$$A_{mm} x_m^k = b_m - A_{m1}\, x_1^k - \ldots - A_{mm-1}\, x_{m-1}^k - A_{mm+1}\, x_{m+1}^{k-1} - \ldots$$

$$\ldots - A_{mn}\, x_n^{k-1}$$

After solving each system of equations, the solution subvectors may be further adjusted by the method of successive over-relaxation or under-relaxation. That is, let $\bar{x}_i^k$ be an arbitrary solution subvector of the kth iteration step. Then define the improved estimate of that solution subvector as

$$x_i^k = x_i^{k-1} + w(\bar{x}_i^k - x_i^{k-1}) \qquad (3.1.4\text{-}6)$$

That is to say, the improved estimate at step k is extrapolated from the Gauss-Siedel estimate and the previous improved estimate. If w=1, the method reduces to that of Gauss-Siedel. The quantity w is called a relaxation parameter, the choice of which determines the rapidity of convergence. For problems where the successive iterations indicate an oscillatory convergence behavior in a predominant number of the elements of the solution subvectors, the rate of convergence will probably be increased by setting w to some value between 0 and 1.0. If the convergence behavior is predominantly overdamped, the rate of convergence may be increased by setting w to some value between 1.0 and 2.0. For a given problem there exists an optimum w such that the number of iterations to convergence is minimized. Several procedures have been suggested for determining this optimum w. Ames(1969) and Bratkovich(1975) provide further information on determining an optimum relaxation parameter. In the SOLN code, the value of w is input by the user as part of the input data.

Convergence of the iteration procedure is defined such that

$$\frac{|x_i^m - x_i^{m-1}|}{|x_i^m|} < \varepsilon \qquad (3.1.4\text{-}7)$$

for each element in each subvector x_i. If the specified maximum number of iterations have been performed and convergence has not been achieved, the last estimates of the solution subvectors may be saved and used as initial guesses for a subsequent series of iterations.

In solving each system of equations in the series of systems described by equations 3.1.4-4 and 3.1.4-5, any one of the direct solution techniques available in the SOLN code may be used. To eliminate redundant calculations in each iteration step, the inverse or decomposition of the diagonal submatrices A_{ii} is computed or read in and stored before the initial guess or iteration procedure begins. Then the matrix multiplications or forward elimination and backward substitution are the only computations made in solving each system of equations involved in the initial guess and subsequent iterations. The decomposition is used with square systems of equations for each diagonal sub-matrix that fits in core at one time. The inverse is used in all other circumstances. This feature significantly reduces the amount of computer time required.

This iteration procedure may also be used to solve a rectangular system of equations. In this case, the Householder procedure must be used to compute the inverses of the diagonal submatrices.

3.2 SYSTEM MODULARITY

The required degree of modularity, which implies that the program should operate independently from and without interferring with other portions of the POTFAN system, has been guaranteed by the use of auxilliary storage devices (tapes, disks, or drums) as the only method by which the various segments of POTFAN can communicate with one another. This has the disadvantage that communicating through auxilliary devices is relatively slow and on some computer systems will require many job control cards to manage the devices. However the disadvantages of this approach are more than offset by the advantages. The principal advantage is that it strictly guarantees segment independence. Also, this approach is necessary to maximize the size of problem that can be solved with a fixed amount of core memory.

3.3 VERSATILITY AND EASE OF USE AND MODIFICATION

These factors are all-important to the usefulness of a program. Although there is no unique formula for guaranteeing them, two important elements of a sufficient formula relevant to the SOLN code have been identified. These are command format programming and the liberal and thoughtful use of comment cards in the source deck.

Command format programming is a phrase coined to describe a programming method which uses words or acronyms (called commands) to control the actions taken by the program. Although this technique is not new, the particular style employed in POTFAN programs originated with the computer program reported in Medan (1973). This technique contributes significantly to the above important factors.

The manner in which command format programs work is the following. First of all, there is a program known as the control program, which may be either the main program or the principal subroutine called by the main program. For the SOLN code the control program is the subroutine SOLNIO. The first two actions performed by the control program are the initialization of default values and the establishment of whether or not the program is being run in a batch or conversational mode. In the conversational mode the program prompts the user for commands and data, and in the event of a recoverable error, pauses to allow the user to perform a fixup. In the batch mode the program echoes each command read in and generally always stops upon detecting an error. Other than these differences, the two modes of operation are identical. Following these actions the program enters the command phase. In this phase the program reads in various four character commands, takes the action associated with the command, and then (provided the command is not STOP) reads in the next command and so on. The specific commands available are given in section 5.2.

Regarding the use of comment cards, there are no specific rules that can guarantee that the inner workings of a program are adequately documented. However, some guidelines have been developed.

The first of these concerns the quantity of comment cards. Some programs that were thought to be adequately commented were examined and it was found that approximately 25 per cent of the source lines were significant comment lines. Most programs are not considered to be well documented unless they contain approximately this ratio of comment lines. Therefore, the SOLN code is documented to this extent.

The second guideline concerns subprogram documentation. The comment cards should state as a minimum the basic purpose of the subprogram and all inputs and outputs except printed output. Each subprogram in the SOLN code was made to comply with this guideline.

In summary, the use of command format programming and careful commenting have made the SOLN code versatile, easy to use, and easy to modify.

3.4 MAXIMIZING PROGRAM SIZE

Two major techniques were used for treating problems involving a large number of components and/or components with large influence matrices. These are the use of auxilliary storage units and dynamic memory allocation.

3.4.1 Auxilliary Storage

The degree to which auxilliary storage units are used in the SOLN code depends on the size and type of problem for which a solution is desired. For systems of equations that can be placed in core all at once, a minimum of auxilliary storage is required. In fact, the influence matrix and right hand sides are placed in auxilliary storage only when they are first read in. When the code automatically determines that the system can be solved entirely in core, the influence matrix and right hand side are read back into core and solved.

For larger systems of equations that cannot fit within core simultaneously, the influence matrix must be partitioned and solved in a piecemeal fashion which requires extensive usage of auxilliary storage units. For problems involving the iteration procedure, auxilliary storage units are used to store each block of the matrix as well as the right hand sides, the current estimate of the solution subvectors, and the previous estimate of the solution subvectors.

This usage of auxilliary storage units allows for the solution of systems of equations whose size is only limited mainly by the amount of auxilliary storage available rather than the amount of core storage available.

3.4.2 Dynamic Memory Allocation

Dynamic memory allocation as used herein refers to packing several small arrays into one large single array. In this way all of the available core space can be utilized.

A necessary procedure in the use of dynamic memory allocation is that dynamically allocated arrays must be passed as variably-dimensioned arrays through subroutine argument lists. This in turn makes the program easy to modify since the storage array into which the smaller arrays are packed needs to be dimensioned only once, e.g., in a small main program.

The storage array in SOLN is denoted as (A). This array is packed differently depending upon the type of calculations being performed. When bulk data is read in, storage is only allocated for that particular set of data. Then the data is placed in temporary storage. The allocation of storage is more complicated at the point where a system of equations is to be solved. Storage in this case is allocated depending upon the method of solution, whether the iteration procedure or the direct procedure is used, and whether the system of equations can be solved in core all at once. In general, however, the arrays that are the length of the number of fragments and the number of rows in the matrix are allocated first. Then the remainder of the available storage is allocated for the influence matrix.

4 PROGRAM DESCRIPTION

To a great extent the description of the inner workings of the program has been relegated to comment cards in the FORTRAN source decks. This includes descriptions of the functions of the subroutines and their input and output. The remainder of this section presents relevant descriptive data which could not effectively be placed on comment cards.

4.1 CALLING STRUCTURE

Figure 4.1-1 shows the subroutine calling structure. Table 4.1-1 shows the calling structure in a different format.

4.2 FLOW CHART

Figure 4.2-1 is a flow chart of the control program SOLNIO.

4.3 COMMON BLOCKS

Table 4.3-1 shows the common blocks used, their sizes, and the subprograms which they appear in.

4.4 LOGICAL UNITS

Table 4.4-1 summarizes the logical units (tape, disks, or drums) which the program uses. Note that not all units would be used for each program run. For the worst case the number of units used would be ten (10). The specific data input from or output onto each of these units except for the line printer unit is discussed in detail in sections 6 and 7.

4.5 MEMORY REQUIREMENTS

Without the working storage array, the SOLN code requires approximately 24000 decimal words of core storage. This requirement includes all system subroutines and internal symbol dictionaries and was determined on the INFONET Univac 1108 operating system without using overlays. The size of the working storage array must be added to this number to determine the total amount of storage required by the program. The memory requirements can be reduced by using overlays.

4.6 RESTRICTIONS AND LIMITATIONS

The most important restrictions regarding the SOLN code are those associated with the limitation in size of the system of equations that can be solved. There are three different types of limitations depending on the manner in which the system is solved. First, there is a limitation to the number of equations that can be solved in core at one time. For a given computer system, this limitation can be determined by the following formula

$$NR = \sqrt{4 + (MXS-3*NIR)} - 2 \qquad \text{(square matrix)}$$

$$NR = \frac{MXS - 3*NIR}{4 + NC} \qquad \text{(non-square matrix)}$$

where NR = Number of equations that can be solved in core at one time.
MXS = Size of the working storage array (A).
NC = Number of unknowns to be determined.
NIR = Number of records of input containing the influence matrix. This is also referred to as the number of influence matrix fragments.

Second, there is a limitation to the number of equations that can be solved even using auxilliary storage devices. This limitation depends upon whether the iteration procedure is used or whether a direct procedure is used.

For a direct procedure the limitation is computed as follows:

$$NR = \frac{MXS-3*NIR+1}{7} \qquad \text{(square matrix)}$$

$$NR = \frac{MXS-3*NIR}{4} \qquad \text{(non-square matrix)}$$

For the iteration procedure the limitation is computed as follows:

$$NRB = \frac{MXS-3*NIR+1}{7}$$

$$NRB = \frac{MXS-3*NIR-MAXVNF}{4}$$

whichever is smaller (square matrix)

$$NRB = \frac{MXS-3*NIR-MAXVNF}{4} \qquad \text{(non-square matrix)}$$

where NRB = The order of the largest diagonal block of the matrix.

MAXVNF = Number of influence coefficients in the largest influence matrix input record.

The third limitation concerns the amount of auxilliary storage available. The temporary storage units INA, INAT, INCBD, KST, INB, INBT, and INVR must each be able to contain the entire influence matrix. Hence, the largest system of equations that can be solved may be limited by the size of the auxilliary storage devices.

5 OPERATING INSTRUCTIONS

The purpose of this section is to provide the user with information necessary to execute the program.

5.1 GENERAL DATA INPUT CONSIDERATIONS

The input data for the equation solver program consists of punched cards and tape, disc, or drum files when the code is utilized in the batch mode, and on-line data with tape, disc, or drum files when it is executed in the conversational mode. All punched card and conversational terminal input is prescribed in NAMELIST or regular formats, whereas all bulk data input through tape, disc, or drum files is unformatted. The latter data is described in sections 6 and 7.

The program is designed to use commands as the basic form of input to control the program flow. These commands consist of four letter words placed in the first four characters of an input record (first four columns of an input card) and are recognized as keys that cause the program to perform particular operations. After the operations are performed, the program flow returns to the beginning of the program and reads the next command. This continues until a STOP command is encountered, whereby the program terminates. Any command input record (card) whose first four characters or columns are left blank is considered a "comment" command. Any command that is not recognized by the program is printed and program flow is either returned to the next command without any operations being performed or terminated depending on the value of the variable CONTIN (see section 5.2.5). The particular commands recognized by the equation solver code are discussed in more detail in section 5.2.

All of the commands available in the equation solver code are related in one way or another to the solution of a system of equations, although several commands in sequence are required to complete all the steps necessary to obtain this solution. Since there are several different solution techniques available in the code, the best method of solution and the associated commands for a given problem will depend upon whether the system of equations (matrix) under consideration is over-determined (rectangular) or not, and whether the system of equations under consideration can be subdivided or blocked effectively so that the iteration procedure can be implemented.

For solving larger systems of equations, the potentially fastest solution technique is the block successive over-relaxation method. This is true whether the system of equations is overdetermined or not. However, the iteration procedure will only converge to a solution if the matrix produced by the system of equations has certain properties. In particular, the matrix must possess submatrix blocks such that the diagonal blocks are more dominant than the off diagonal blocks. The more dominant the diagonal blocks, the faster the iteration procedure will converge to a solution.

Matrices that do not possess this property are best solved by the direct techniques. For square matrices, the LU decomposition method is the fastest of the direct methods, and for rectangular matrices the Householder procedure should be used. In the event that the data defining the system of equations under consideration requires more computer core storage than is available, the equation solver code automatically places the data in temporary storage and determines the solution to the system of equations by the procedures for large systems as discussed in Section 3.3.

5.2 INPUT DESCRIPTION

The first line of input to the program consists of a single logical variable. Its value depends on whether the command inputs are to be made utilizing the conversational or batch modes, .TRUE. for conversational, .FALSE. for batch mode. The format for this input record (card) is L1.

The remaining input for the equation solver code consists mainly of operation commands, input file names, and option flags. A command statement must begin in the first column or with the first character of an input card or record. Only the first four characters of any command are recognized by the program. If the first four characters or columns are left blank, the command is assumed to be for comment purposes, and the entire card or line is printed. In the conversational mode, an unrecognizable command returns control to the next command after the unrecognizable characters are printed; otherwise, the program terminates unless the variable CONTIN has been set to .TRUE. (see DATA command. A detailed description of the commands recognized by the code, as well as the input associated with them, is given below. All logical unit numbers may range from 1 to 99, except 5 and 6. File identification numbers may range from 1 to 9999. See Appendix A for an explanation of file identification numbers.

5.2.1 (4-Blanks) Command

Explanation

Any command with blanks in the first four positions is considered a comment card. All information on this card or input record is printed, then control proceeds to the next command.

5.2.2 BCRE Command

Explanation

This command causes the boundary condition constant vector files to be read in and placed in temporary storage. This boundary condition constant vector file is ordinarily created by the boundary condition program. This command should not be given before the CNTL command (when required) or the VNRE command, and must precede the SOLV, RIVNS, or MULT commands.

Input

(NTBCR,NBCR(I),I=1,NCMP)--The identification number and logical unit number of each boundary condition constant vector file.

Format

2I5

Comments

Enter one identification number and one logical unit number per line or card of input. The input sequence must be consistent with the input sequence of the input influence submatrices (see VNRE command).

5.2.3 CNTL Command

Explanation

This command defines the number of aircraft components and the overall type of solution procedure to be used in solving the system of equations under consideration. It is the first command given if the internal logical units do not require initializing with the DATA command. This command must precede all other commands except blank and DATA commands except when there is only a single component whose influence matrix can fit in core. In the latter case, the CNTL command is not required.

Input

NCMP--Number of aircraft components associated with the system of equations under consideration. This determines the number of boundary condition and influence matrix files that must be read in.

ITERAT--Flag which designates whether an iteration procedure or a direct solution procedure is to be used to solve this system of equations.

ITERAT =
0--Direct solution
1--Iteration solution

Format

2I5

5.2.4 COMB Command

Explanation

This command is used only when the system of equations under consideration is too large for available core storage and the direct solution procedures are being used to solve the system. It combines all submatrices read in by the VNRE command into one matrix which is stored in column length records on a temporary storage device. The command need not be used unless both circumstances mentioned above exist. No further data input is required with this command. If used, it must immediately follow the VNRE command.

5.2.5 DATA Command

Explanation

The DATA command is used to initialize any or all internal logical units used by the program. This includes any temporary storage devices as well as the input and output logical units. The DATA command is also used to set the variable CONTIN. This command need not be used if the default values of the variables are satisfactory. This command, if needed, must precede all of the commands except the blank command.

Input

NTCP--Logical unit number of the output device used for conversational mode operations. The default value is 6.

NTP--Logical unit number of the output device used for batch mode operations. The default value is 6.

NTCR--Logical unit number of the input device used for conversational mode operations. The default value is 5.

NTRR--Logical unit number of the input device used for batch mode operations. The default value is 5. The value of NTR is set equal to NTRR immediately following the read statement.

INA--Logical unit number for a temporary storage device. The default value is 11. However, during a COMB command, it may be changed to 12. See section 7.4 for an explanation of the temporary storage devices.

INAT--Logical unit number for a temporary storage device. The default value is 12. However, during a COMB command, it may be changed to 11.

INCBD--Logical unit number for a temporary storage device. The default value is 14.

KST--Logical unit number for a temporary storage device. The default value is 13.

INB--Logical unit number for a temporary storage device. The default value is 15.

INBT--Logical unit number for a temporary storage device. The default value is 17.

INVR--Logical unit number for a temporary storage device. The default value is 18.

NTIR--Logical unit number for a temporary storage device. The default value is 16.

CONTIN--Logical variable governing program action upon encountering an unrecognizable command. Unless CONTIN is .TRUE., execution will terminate with an unrecognizable command. The default value for CONTIN is .TRUE. under conversational operation and .FALSE. under batch operation.

Format
Namelist MDATA.

5.2.6 INVE Command

Explanation
This command causes the inverse of a matrix to be computed and stored on a tape, disc, or drum file. The user must input the identification number and logical unit number of the tape, disc, or drum file where this inverse is to be stored. In addition, the user must provide a title for the file. A special option with this command allows the user to compute and store the decomposition of a matrix instead of the inverse (see Section 3.1). The only restrictions for

the decomposition option are that the matrix to be decomposed must fit in core all at once and that it must be a square matrix. The INVE command must follow the VNRE and (if needed) COMB commands.

Input

METHOD--Option flag which designates which solution technique is to be used to create the inverse or decomposition.

METHOD =

0--LU decomposition method (square matrix only)

1--Householders' technique (rectangular or square matrix)

NTINVW--Identification number of tape, disc, or drum file where the computed inverse is to be stored. The logical unit number associated with this file is INVR which may be input with the DATA command at the beginning of each test case.

IDC--Option flag which designates whether the inverse or the decomposition of the matrix is to be computed.

IDC =

0--Inverse

1--Decomposition

NTITL--Number of words in the title describing the inverse or decomposition file. NTITL must be greater than zero but not exceed 100.

(TITL(I), I=1,NTITL)--A one dimensional alphanumeric array that may be used to describe the contents of the inverse file.

Format

3I5/I5/(20A4)

5.2.7 IVNR Command

Explanation

This command causes an inverse or decomposed matrix file to be read in and placed in temporary storage and is a sequel to the INVE command. The intent of this command is to allow the user to read in an inverse without computing it, then multiply the inverse by any number of constant vectors to obtain the solution vectors. In the case of a decomposed matrix, a forward elimination and backward substitution are performed on the decomposed matrix and constant vectors to obtain the solutions. The IVNR command, however, only reads the inverse or decomposed matrix. Other commands read the constant vectors (BCRE command) and perform the multiplication or forward elimination and backward substitution (MULT command). The IVNR command must follow a CNTL command and a BCRE command and precede a MULT command.

Input
NTINVR--Identification number of the file where the inverse or decomposed matrix is stored.

NTINV--Logical unit number of the file where the inverse or decomposed matrix is stored. If entered as zero, the program uses 18 instead.

Format
2I5

5.2.8 MULT Command

Explanation
This command is used in conjunction with the IVNR command. It causes an inverse matrix to be multiplied by any number of constant vectors or it causes a forward elimination and backward substitution to be performed on a decomposed matrix and any number of constant vectors. The information stored on the file that was read by the IVNR command indicates whether an inverse or a decomposed matrix existed on the file and the code automatically performs the appropriate computations to obtain the solution vectors. The IVNR and BCRE commands must precede the MULT command.

Input
(OUTA(I),OUTANT(I),I=1,NCMP)--One dimensional integer arrays defining the identification numbers and logical unit numbers of the files where the computed solution vectors are to be stored.

Format
2I5

5.2.9 PART Command

Explanation
This command causes the program to partition the matrix that was read in by the VNRE command and combined, if necessary, by the COMB command. It is a utility command that need not ordinarily be used unless it is desirable to partition a matrix into submatrices in a manner that is different from the way the matrix is already partitioned. The PART command will cause the matrix to be subdivided into equal sized square blocks, if possible. If this is not possible, then the blocks will be made as equal in size as is possible. This command must follow the VNRE and (if used) COMB commands and precede the SOLV and INVE commands.

Input

NPA--The number of partitions that a row or column of the matrix is to be subdivided into. The number of partitions in the matrix will be NPA**2.

(NP(I),I=1,NPA**2)--A one dimensional array defining the logical unit numbers of the files where each submatrix partition is to be stored. A total of NPA**2 entries must be made. Place the values for one row of partitions on each line or card.

Format

I5/(16I5)

5.2.10 SOLV Command

This command causes the program to compute the solution to the system of equations as defined by the VNRE and BCRE commands. The user need not define a solution technique for solving the system of equations; the code will automatically determine this for him. However, if he chooses to override the automatically chosen technique, he may do so. If the user chooses an inappropriate technique for a given system, the code will automatically choose the best technique and override his decision. This may occur if the user attempts to solve a rectangular matrix using a technique for square matrices. The various solution techniques available in the code were discussed in detail in Section 3. If the iteration procedure option is in effect, this command defines the method for solving the system of equations associated with each diagonal submatrix of the blocked matrix. The SOLV command must follow the VNRE, COMB (if used), and BCRE commands. If a user input solution technique is used, the calling sequence for this procedure should be placed in the subroutine INVRS at the designated location.

If the iteration procedure is in effect and is non-convergent, the last estimate of the solution subvectors will be printed and stored on files. The iteration procedure may be restarted by reading these final approximations into the program as initial guesses at the solution subvectors instead of have them computed. In fact, any predetermined solution subvectors may be read in as an initial guess for the iteration procedure instead of having them computed when the iteration option is in effect. In other words this is the number components for which an initial guess of the solution will be supplied.

Input

METHOD--Option flag defining which solution technique will be used in solving the system of equations.

METHOD =

0--LU decomposition method
1--Householders' method
2--User input method

IPS--The number of predetermined solutions to be read when the iteration option is in effect. In other words this is the number components for which an initial guess of the solution will be supplied.

(OUTA(I),OUTANT(I),I=1,NCMP)--Two one dimensional arrays defining the identification number and logical unit number where the solution subvectors of the system of equations under consideration are to be stored. One pair of numbers is required for each aircraft component.

IPRINT--A print parameter that prescribes whether the solution vector is printed out.
IPRINT=0--Solution vector is printed.
=1--Solution vector is not printed.

(I,INPS(I),INPSNT(I),J=1,IPS)--A simple variable and two one dimensional arrays that indicate the aircraft component sequence number, the identification number, and logical unit number of any predetermined solution subvector files to be used as an initial guess to the solution for the iteration procedure. This input is only required ir the iteration procedure option is in effect. If IPS is zero, this information must not be input. The solution subvectors on these files will be used as the initial guess for the iteration procedure instead of having the initial guesses computed. One set of these numbers is required for each aircraft component for which an initial guess is to be input.

MAXIT1--The maximum number of iterations allowed for the iteration procedure to converge. This is only required if the iteration procedure option is in effect. If MAXIT1=0, only the initial guess will be computed, printed, and stored.

OMEGA--Relaxation factor that alters the convergence characteristics of the iteration solution.

0<OMEGA<1 (under-relaxation)
1<OMEGA<2 (over-relaxation)

The default value is 1.0. This input is only required when the iteration solution option is in effect.

ITERP--Print parameter that defines the amount of intermediate results to be printed during the iteration solution procedure.

ITERP =

0--No intermediate results will be printed.

Not-zero--Intermediate results will be printed every ITERP iterations.

The intermediate results consist of the influence of each submatrix block on every other submatrix block, the constant vectors, and the most recent estimate of the solution. If ITERP ≠ 0, these quantities are also printed for the initial guess at the solution. This input is only required if the iteration procedure option is in effect.

Format

(2I5)--For METHOD, IPS, OUTA(I), OUTANT(I), IPRINT.

(3I5)--For I, INPS(I), INPSNT(I).

(I5/F5.2/I5)--For MAXIT1, OMEGA, ITERP.

Comments

The inverse or decomposition of the diagonal submatrices associated with the iteration procedure may also be read in instead of being computed. This is discussed in the VNRE command.

5.2.11 STOP Command

Explanation

This command terminates the program. No further operations can follow the STOP command.

5.2.12 VNRE Command

Explanation

This command reads the submatrices making up the influence coefficients for the system of equations under consideration and places them in temporary storage. The matrix is usually subdivided into submatrices that are placed on separate tape, disc, or drum files. For potential flow problems, these submatrices usually represent the influence of a single aircraft component on another single aircraft component. If the iteration procedure is in effect, each submatrix also represents a block of the block iteration procedure. A special option is available with this command when the iteration option is in effect. It allows the user to avoid calculating the inverse or decomposition of any one of the diagonal submatrix blocks that would ordinarily be computed during the iteration procedure. Instead, the user may read in the inverse or decomposition itself. For those diagonal submatrices that are read in the ordinary manner, the inverse or decomposition is computed during the iteration procedure calculations. The VNRE command must precede the BCRE command and the COMB command (if used) and must also follow the CNTL and DATA commands.

Input

NTVNR,NTVN--The identification number and logical unit number of a single submatrix of the influence matrix file. One pair of numbers is entered per line. The submatrices must be entered in rowwise order. That is, referring to the figure in Section 3.1.4, the submatrices must be entered in the following order: A_{11} ,...,A_{1n} ,A_{21} ,...,A_{2n} , ...,A_{n1} ,...,A_{nn} .

INVF(I)--Option flag that designates whether the associated file contains a submatrix or whether it contains the inverse or decomposition of that submatrix. This is only input with the diagonal submatrices and is entered on the same line as the logical unit number or identification number for that submatrix.

INVF(I) =
0--Submatrix.
-1--Inverse or decomposition of the submatrix.

Format

2I5 (for off-diagonal submatrices)
3I5 (for diagonal submatrices)

5.3 SYSTEM CONTROL CARDS

This section describes the control cards that are necessary to run the SOLN code on the various computer systems that have been or are being used to run the SOLN code.

5.3.1 Infonet Univac 1108 System

Since this system allows automatic file definition commands determined from the file identification numbers (see Appendix A), the only control card required is the program name, SOLN/PCTF. All files created except scratch files will automatically show up in the user's catalog. The names of all created files will be identical to the input file identification numbers. Also each created file will have a version identifier that defines the type of file that it is. The version identifier for solution files is SO; for inverse files it is IN. All files created will reside in the LIB$ library. Thus, for example, a solution file created with an identification number of 1023 will be assigned the name 1023.SO/LIB$.

6 PROGRAM INPUT BINARY FILES

The SOLN code uses two types of files created by other POTFAN segments. These files contain the coefficient matrix (influence matrix files) and the right hand sides (boundary condition files). These files are accessed according to the procedures in Appendix A and Section 5.3. Each of these files conforms to the standardized POTFAN format, which is discussed in Appendix B. The notation used in the remainder of this section will be clear to the reader if he reads Appendix B. All of the data available on these files is not used by SOLN. The following subsections describe the files briefly and which quantities affect the SOLN code. The program may also input an inverse file, a decomposition file, or a solution file that was previously computed by the SOLN code. These files are discussed in Section 7.

6.1 INFLUENCE MATRIX FILES

Each influence matrix file contains all the information necessary to describe the boundary condition influence of one aircraft component on some other aircraft component. The file itself consists of two or more records of information. The first record is referred to as the introductory record, while all subsequent records contain the influence coefficients. In many cases the influence matrix is very large and cannot be input or output all at one time on one record of information. Therefore, several records are often required to input or output the entire influence matrix file. Each record of influence coefficients is referred to as a matrix fragment. Each influence matrix file is referred to as a submatrix. Many influence matrix files (submatrices) may be required to completely define the entire influence matrix associated with a given flow field problem consisting of several aircraft components.

The following information from the influence matrix files is used directly by the SOLN code. The variables marked with an asterisk are necessary for proper execution of the program. All variables without an asterisk are informative only and are not necessary for proper execution of the SOLN program but are transferred to the solution files to be used as input for subsequent programs in the POTFAN system.

1st Record

*NCTIME--Number of words in the data and time array.

(CTIME(NCTIME))--Array containing the date and time of creation of the influence matrix file.

*NTITL--Number of words in the title of a self-influencing submatrix file.

(TITL(NTITL))--Title of a self-influencing submatrix file.

*NRECS--One plus the number of submatrix fragments.

ID(1)--Identification number of a self-influencing submatrix geometry file.

ID(3)--Identification number of unconstrained self-influence submatrix.

ID(5)--Identification number of constrained self-influence submatrix.

*NLOG

LOG(2)=TOP--Flag indicating whether an upper surface boundary condition has been applied.

LOG(3)=BOT--Flag indicating whether a lower surface boundary condition has been applied.

LOG(11)=DBLT--Flag indicating whether doublet type singularities have been used to represent an aircraft component.

LOG(12)=SOURCE--Flag indicating whether source type singularities have been used to represent an aircraft component.

LOG(13)=CNSTRN--Flag indicating whether the influence matrix is constrained.

*NINT

*INT(8)=NSING--Number of unknowns associated with the self-influence matrix file.

INT(12)=ICTYPE--Not used anymore.

INT(14)=NFC--Number of constraint function coefficients representing the singularity distribution of a self-influence submatrix file.

INT(16)=NCFFX--Number of constraint function fragments in the N1 direction.

INT(17)=NCFFY--Number of constraint function fragments in the N2 direction.

INT(18)=ICV--Flag indicating the type of constraint variables used to generate the constraint function.

NFLT

*FLT(1)=XMACH--Mach number. The SOLN code checks all influence matrix files to see that they correspond to the same Mach number. If there is an inconsistency, the program will stop and print a diagnostic message.

2nd Record and Subsequent Records

*J1=NC--Number of columns in the submatrix.

*J2=NR--Number of rows of the submatrix on this record.

J3=1

*NW=NC*NR--Number of influence coefficients on this record.

*(A(I),I=1,NN)--Influence coefficients on this record. These are stored columnwise. When multiple records are required to store the submatrix, then a certain number of complete rows of influence coefficients are contained on each record but are still stored columnwise.

6.2 BOUNDARY CONDITION FILES

Each boundary condition file contains all the information necessary to describe the influence of the uniform and nonuniform freestream and the rotation rates on the boundary conditions of a single aircraft component. The file itself consists of three or more records of information. The first record is referred to as the introductory record. Each group of two records thereafter describes all the flow field parameters associated with a single set of boundary conditions as well as their influence on this aircraft component. There may be several sets of boundary conditions on a single boundary condition file. The second, fourth, sixth, etc., records on the file contain the right hand side subvector associated with the system of equations describing the specified flow boundary condition. The third, fifth, seventh, ninth, etc., records contain the parameters defining the freestream conditions; that is, the freestream velocity vector, the angles of attack and sideslip, the rotation rates, and the center of gravity location for the flow field.

The following information from the boundary condition files is used by the SOLN code. The variables marked with an asterisk are necessary for proper execution of the program. All variables without an asterisk are informative only and are not necessary for proper execution of the SOLN program but are transferred to the solution files to be used as input for subsequent programs in the POTFAN system. asterisk.

<u>1st Record</u>

*NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the date and time of creation of the boundary condition file.

*NTITL--Number of words in the title.

(TITL(NTITL))--Title of the boundary condition file.

*NRECS--One plus two times the number of sets of boundary conditions.

ID(1)--Identification number of the geometry file associated with this aircraft component.

ID(2)--Identification number of the boundary condition file.

LOG(5)=DBLT--Flag indicating whether doublet type singularities have been used to represent an aircraft component.

LOG(6)=SOURCE--Flag indicating whether source type singularities have been used to represent an aircraft component.

LOG(7)=TOP--Flag indicating whether an upper surface boundary condition has been applied.

LOG(8)=BOT--Flag indicating whether a lower surface boundary condition has been applied.

*INT(2)=NROW--Number of rows.

*INT(3)=NSETS--Number of sets of boundary conditions on this file.

2nd, 4th, 6th, . . . Records

*J1=NROW--Number of elements in the constant subvector.

J2=1

J3=1

NW=NROW--See above.

*(A(NROW))--Elements of the constant subvector.

3rd, 5th, 7th, . . . Records

J1=11

J2=1

J3=1

NW=11

A(NW)--Array containing the freestream velocity vector, the angles of attack and sideslip, the rotation rate vector, and the center of rotation location for this set of boundary conditions.

7 PROGRAM OUTPUT

Output from the program consists of line printer output, data left on scratch files, and various POTFAN files. The line printer output is meant to be self explanatory and will not be discussed further. The POTFAN files that are created are created according to the procedure in Appendix A and conform to the format in Appendix B. Control cards for managing the scratch and POTFAN files are given in Section 5.3. The POTFAN and scratch files are discussed in more detail in the following subsections.

7.1 SOLUTION FILES

Each solution file contains all the information relevant to that portion of the solution vector associated with a single aircraft component. Each such portion of the solution vector is referred to as a solution subvector. The file itself consists of three or more records. The first record is the introductory record. The second record contains the parameters defining the freestream conditions for all sets of boundary conditions for which a solution was computed; that is, the freestream velocity vector, the angles of attack and sideslip, the rotation rates, and the center of rotation location for the flow field. The third and subsequent records contain the solution subvectors, one record for each boundary condition.

The following information is written on each solution file.

1st Record

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the date and time that the solution file was created.

NTITL--Number of words in the title array.

(TITL(NTITL))--Title of the geometry file for this aircraft component.

NRECS--Two plus the number of sets of solution subvectors.

(IFCRM(NRECS))--0, 1, 1, . . .

NID=5

ID(1)--Identification number of the geometry file associated with this aircraft component.

ID(2)--Identification number of the self-influence submatrix file associated with this aircraft component.

ID(3)--Identification number of the inverse file used to generate this solution subvector, if such a file was used. Otherwise ID(3) is zero.

ID(4)--Identification number of the boundary condition file associated with this aircraft component.

ID(5)--Identification number of this solution file.

NLOG=5

LOG(1)=DBLT--Flag indicating whether doublet-type singularities have been used to represent an aircraft component.

LOG(2)=SOURCE--Flag indicating whether source type singularities have been used to represent an aircraft component.

LOG(3)=TOP--Flag indicating whether an upper surface boundary condition has been applied.

LOG(4)=BOT--Flag indicating whether a lower surface boundary condition has been applied.

LOG(5)=CNSTRN--Flag indicating whether the influence matrix is constrained.

NINT=8

INT(1)=ICTYPE--Not used at present.

INT(2)--number of elements in the solution subvector (i.e., number of unknowns).

INT(3)--Number of sets of boundary conditions.

INT(4)=METHOD--Method of solution.

INT(5)=NSING--Number of elements in the solution vector if it were unconstrained.

INT(6)=NCFFX--Number of constraint function fragments in the N1 direction (if the matrix is constrained).

INT(7)=NCFFY--Number of constraint function fragments in the N2 direction (if the matrix is constrained).

INT(8)=ICV--Flag indicating the type of constraint variables used to generate the constraint function.

NFLT=1

FLT(1)=XMACH--Mach number.

2nd Record

J1=NSOL-Number of solutions.

J2=11

J3=1

NW=11*NSOL

(UINF(NSOL)),(VINF(NSOL)),(WINF(NSOL))--Components of the unit freestream velocity vector for each set of boundary conditions.

(ALPHA(NSOL))--Angle of attack for each set of boundary conditions.

(BETA(NSOL))--Angle of sideslip for each set of boundary conditions.

(P(NSOL)),(Q(NSOL)),(RR(NSOL))--Components of the freestream rotation rate vector for each set of boundary conditions.

(RCGX(NSOL)),(RCGY(NSOL)),(RCGZ(NSOL))--Components of the position vector to the center of rotation of the flow field for each set of boundary conditions.

3rd and Subsequent Records

J1=NR--Number of elements in the solution subvector.

J2=1

J3=1

NW=NR--See above.

(X(NR))--Elements of the solution subvector for a particular set of boundary conditions.

7.2 INVERSE MATRIX FILES

Each inverse matrix file contains all the information pertaining to the inverse of a matrix that was assembled and computed from one or more influence matrix files. That is, the system of equations from which this inverse matrix was generated may have involved one or more aircraft components. The file itself consists of three or more records. The first record is the introductory record. The second record contains all of the information from the influence matrix files that must be transferred to the solution files when the inverse is multiplied by a constant vector to produce a solution vector. The third and subsequent records contain the inverse matrix coefficients. Since the storage required to output the entire inverse matrix on a single record may be more than is available, several records may be necessary to output the complete matrix.

The following information is written on an inverse matrix file.

<u>1st Record</u>

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the date and time that the inverse matrix file was created.

NTITL--Number of words in the title array.

(TITL(NTITL))--Title of the inverse matrix file.

NRECS=Two plus the number of records required to contain the entire inverse matrix.

(IFORM(NRECS))=0,0,1,1,...

NID=1

ID(1)=Identification number of the inverse matrix file.

NLOG=1

LOG(1)=FIT--Flag indicating whether the matrix fits in core all at once.

NINT=7

INT(1)=KM--Number of rows in the inverse matrix.

INT(2)=NCPF--Maximum number of columns in each inverse matrix output record.

INT(3)=NCMPS--Number of aircraft components associated with the system of equations from which the inverse matrix was created.

INT(4)=INV--Method used to create the inverse.

INT(5)=NCFC--Number of records required to output the entire inverse matrix. This is also referred to as the number of inverse matrix fragments.

INT(6)=IDC--Flag indicating whether this matrix is an inverse or an LU decomposition matrix. A value of 0 indicates an inverse, while a value of 1 indicates an LU decomposition.

INT(7)=NCOL--Number of columns in the inverse matrix.

NFLT=1

FLT(1)=XMACH--Mach number.

2nd Record

J1=NCMPS--See above.

J2=7

J3=1

NW=7*NCMPS

(ICNST(NCMPS))--Array indicating whether the singularity distribution associated with each aircraft component is constrained or not.

(ICT(NCMPS))--Not used at present

(NCFFXA(NCMPS))--Array indicating the number of N1 direction constraint function fragments associated with each aircraft component.

(NCFFYA(NCMPS))--Array indicating the number of N2 direction constraint function fragments associated with each aircraft component.

(ICVA(NCMPS))--Array indicating the type of constraint variables used with the constraint function associated with each aircraft component.

(IDA(NCMPS))--Identification number of the self-influence submatrix associated with each aircraft component.

(NNC(NCMPS))--Ending column number of the self-influence submatrix associated with each aircraft component as positioned in the entire influence matrix from which the inverse was computed.

3rd and Subsequent Records

J1=KM--Number of rows in the inverse matrix.

J2=NCPF--Number of columns in this fragment of the inverse matrix.

J3=1

NW=J1*J2--Number of inverse matrix coefficients in this record.

(A(NW))--Inverse matrix coefficients for this fragment.

7.3 LU DECOMPOSITION MATRIX FILES

Each LU decomposition matrix file contains all the information pertaining to the LU decomposition of a matrix that was assembled from one or more influence matrix files. An LU decomposition matrix is analogous to an inverse matrix in that it is determined without knowledge of the constant vector. Hence, any number of constant vectors may be solved using the same LU decomposition matrix. The system of equations from which this LU decomposition matrix was generated may have involved one or more aircraft components. The file itself consists of four records. The first two records are exactly the same as the first two records of the inverse files (Section 7.2). The third record contains the permutation vector associated with the LU decomposition matrix which tells how the row interchanges took place in the decomposition. The fourth record contains the LU decomposition matrix coefficients. At present, an LU decomposition matrix may be computed by the SOLN code only if the entire matrix fits in core at one time.

The following information is written on an LU decomposition file.

1st Record

Same as 1st record of inverse file except that IFORM(3) is 1. See Section 7.2.

2nd Record

Same as 2nd record of inverse file. See Section 7.2.

3rd Record

J1=KM--Number of rows in the LU decomposition matrix.

J2=1

J3=1

NW=KM--See above.

(IP(NW))--Elements of the permutation vector.

4th Record

J1=KM--Number of rows in the LU decomposition matrix.

J2=NCPF--Number of columns in the LU decomposition matrix.

J3=1

NW=J1*J2--Number of coefficients in the LU decomposition matrix.

(A(NW))--Coefficients of the LU decomposition matrix.

7.4 SCRATCH FILES

A brief description of each scratch file is given below as well as an indication of the circumstances under which each file is used.

INA--This file is used in a variety of ways. In general, it is needed in those circumstances where the influence matrix, inverse matrix, or decomposition matrix does not fit in core at one time. In addition, it is used to store the solution sub-vectors in the iteration procedure and the matrix partitions created during a PART command. The default logical unit number is 11.

INAT--This file is used in tandem with the file INA. That is, information on these files are manipulated and stored by transferring the information back and forth from INA to INAT and vice versa. It is needed in the same circumstances that require the use of INA. The default logical unit number is 12.

KST--This file is used to store the fragments of the influence matrix as read in from the POTFAN influence matrix files. This file is required for any cases using an RVNS command. The default logical unit number is 13.

INCBD--This file is used to store the partitions of the entire influence matrix in those situations where the influence matrix does not fit in core at once and a direct solution procedure is desired. In addition, this file is used to temporarily store the right hand side vectors. Hence, the file is required for any cases using an RBCS command or where the direct solution of a larger-than-core square matrix is desired. The default logical unit number is 14.

INB--This file is used during inversion of a matrix that does not fit in core at one time. Individual blocks of the inverse are temporarily stored on INB. This file is also used to manipulate the right hand side vectors and place them in the appropriate order. Hence. the INB file is required when inversion of large square matrices are performed and whenever the RBCS command is used. The default logical unit number is 15.

INBT--This file is used in tandem with the file INB. That is, information on these files are manipulated and stored by transferring the information back and forth. It is needed in the same circumstances that require the use of INB. The default logical unit number is 17.

INVR--This file is used as a temporary storage device when the iteration procedure option is in effect. Under these circumstances, the fragments of the influence matrix are temporarily placed on this file. Otherwise, INVR is used as the logical unit for any POTFAN inverse files that are output from the SOLN code. The default logical unit number is 18.

8 TEST CASES

This section describes a comprehensive series of test cases which were developed to thoroughly test the program. These test cases are very useful as an aid in familiarizing the user with the manner in which the SOLN code operates. They may also be used as a debugging aid when transferring the SOLN code to a different computer system.

A total of seven test cases are included in this section. Only two different systems of equations were used to perform these test cases; however, a different method of solution was used in each case.

The first system of equations produces a square influence matrix of order ten. An illustration of this system of equations is shown in Figure 8.0-1. A total of three aircraft components are associated with this system of equations. Therefore, the matrix is subdivided into nine submatrices and three constant subvectors. The constant subvectors contain 5, 3, and 2 elements, respectively. Each submatrix and constant subvector are placed on a separate file. The matrix coefficients associated with each submatrix file were stored on two or more records. A total of twenty-two records were used to store the entire influence matrix. Two sets of boundary conditions were placed on each constant subvector file. The solution to the first set of conditions consists of the integers from one through ten. The solution to the second set consists of every other integer from two through twenty. The formulae used to generate the matrix coefficients and constant vector elements are the following:

$$a_{ij} = (n - |i-j|)^7 \qquad (8.0-1)$$

$$b_i = \sum_{j=1}^{n} a_{ij}\, j \qquad \text{(1st set)} \qquad (8.0-2)$$

$$b_i = \sum_{j=1}^{n} 2a_{ij}\ j \qquad \text{(2nd set)} \qquad (8.0\text{-}3)$$

The matrix produced by the above formulation is diagonally dominant and amenable to iteration techniques. This system of equations was used in test cases 1, 3, 4 and 5.

The second system of equations is overspecified and hence produces a rectangular influence matrix. In this case the matrix had seventeen (17) rows and ten (10) columns. An illustration of this system of equations is shown in Figure 8.0-2. Three aircraft components are also associated with this system of equations. Therefore, the matrix is subdivided into nine submatrices and three constant subvectors. The constant subvectors contain 7, 6, and 4 elements, respectively while the solution subvectors contain 5, 3, and 2 elements, respectively. As with the first system of equations, a total of twenty-two (22) records were used to store the entire influence matrix, and two sets of boundary conditions were placed on each constant subvector file. The solutions to both sets of conditions are the same as those for the first system of equations. The formulae used to generate the matrix coefficients and constant vector elements are the following:

$$a_{ij} = (n - |i-y|)^7 \qquad (8.0\text{-}4)$$

$$b_i = \sum_{j=1}^{m} a_{ij}\ j \qquad \text{(1st set)} \qquad (8.0\text{-}5)$$

$$b_i = \sum_{j=1}^{m} 2a_{ij}\ j \qquad \text{(2nd set)} \qquad (8.0\text{-}6)$$

where

$$y = c_0 + c_1 i + c_2 i^2 \qquad (8.0\text{-}7)$$

n = number of rows in the matrix

m = number of columns in the matrix

The coefficients c_0 , c_1 , c_2 are determined by satisfying in a least squares sense the rectangular system of equations that is generated by setting y=j at the location of the first and last element of each diagonal submatrix. This non-simple equation for y is required in order to achieve a system that can be solved by iteration. This system of equations was used for test cases 2, 6 and 7.

The computer programs used to generate the two systems of equations described above are presented in Figs. 8.0-3 and 8.0-4. The input used to run these programs and place the results on the various files is shown in Figs. 8.0-5 and 8.0-6.

8.1 TEST CASE NO. 1

The first test case solves the first system of equations by direct LU decomposition. It is assumed that the entire influence matrix will fit in core at one time. This situation causes the fastest direct solution technique to be used to solve the system. The input for this case is shown in Fig. 8.1-1 and the output is shown in Fig. 8.1-2.

8.2 TEST CASE NO. 2

The second test case solves the second system of equations by the direct Householder procedure since it is a rectangular matrix. The LU decomposition technique could not solve this system of equations. It is assumed that the entire influence matrix will not fit in core at one time, but that the largest influence matrix fragment will fit in core. Hence, the COMB command is used to combine all of the influence matrix fragments into a single matrix and place the matrix in temporary storage. The extended Householder procedure for matrices too large to fit in core at one time is automatically used to solve the system. The input for this case is shown in Fig. 8.2-1 and the output is shown in Fig. 8.2-2.

8.3 TEST CASE NO. 3

The third test case solves the first system of equations again by direct LU decomposition. However, it illustrates the use of a different series of commands that may be advantageous for situations where the same influence matrix is used on several different occasions in conjunction with different constant vectors. That is, the matrix decomposition is created and stored on a file for later use. This decomposition is the most time consuming portion of the LU decomposition procedure and is independent of the constant vector. Later, when the solution to a particular system of equations is desired, the decomposition file and constant vector files are read in and only the forward elimination and backward substitution are performed to determine the solution. It is assumed that the entire influence matrix fits in core at one time. In fact, the SOLN code at present only possesses the capability for creating and storing a decomposed matrix if the influence matrix fits in core at one time. The input for this case is shown in Fig. 8.3-1 and the output is shown in Fig. 8.3-2.

8.4 TEST CASE NO. 4

The fourth test case solves the first system of equations utilizing the block successive over-relaxation iteration procedure. Some of the additional iteration procedure options are also exercised in order to illustrate some of the capabilities available with this procedure. The diagonal submatrices are solved by LU decomposition. The decomposition of one of the diagonal submatrices is assumed to have been computed and stored at some previous time. Hence, instead of being computed during the course of the iteration procedure, it need only be read in. In addition, it is assumed that an approximate solution has been previously obtained for one of the aircraft components. Hence, it need also only be read in as an initial guess at the solution rather than having an initial guess computed for that aircraft component during the course of the iteration procedure. A maximum of ten iterations are allowed for convergence. A mild amount of over-relaxation is used to accelerate convergence. Printouts of some intermediate iterations are also specified. The input for this case is shown in Fig. 8.4-1 and the output is shown in Fig. 8.4-2.

8.5 TEST CASE NO. 5

The fifth test case solves the first system of equations by a two step combination of techniques. In the first step, the solutions for the first two aircraft components are determined as though there were no third aircraft component. That is, only the submatrices and constant subvectors associated with the first two aircraft components were read in and solved. Then the simultaneous solutions for all three aircraft components were determined by the block successive over-relaxation iteration procedure with the solutions from the first step input as initial guesses to the solutions for the first two aircraft components. The influence on the third aircraft component of the first two and vice versa is thus determined iteratively. This situation is analogous to the problem of determining the effect of wind tunnel walls on the characteristics of an aircraft. That is, initially a free-air solution is obtained for the aircraft and then a solution is obtained with the aircraft in the wind tunnel using iteration. The input for this case is shown in Fig. 8.5-1 and the output is shown in Fig. 8.5-2.

8.6 TEST CASE NO. 6

The sixth test case solves the second system of equations in a manner similar to that used to perform the fourth test case. That is, the system is solved by the block successive over-relaxation iteration technique. However, since the second system of equations is rectangular rather than square, the diagonal submatrices are solved by the Householder procedure, and the inverse of one of the diagonal submatrices is read in instead of the decomposition of the matrix. The input for this case is shown in Fig. 8.6-1 and the output is shown in Fig. 8.6-2.

8.7 TEST CASE NO. 7

The seventh test case solves the second system of equations in a manner similar to that used to perform the fifth test case. That is, the system is solved by a two step combination of techniques. However, since the second system of equations is rectangular rather than square, the Householder procedure was used to determine the solutions for the first two aircraft components and also to solve the diagonal submatrices during the iteration procedure. The input for this case is shown in Fig. 8.7-1 and the output is shown in Fig. 8.7-2.

9 REFERENCES

Ames, William F. (1969). Numerical Methods for Partial Differential Equations. New York: Barnes and Nobles.

Bratkovich, A., and Marshall, F. J. (1975). "Iterative Techniques for the Solution of large Linear Systems in Computational Aerodynamics," J. Aircraft 12, 2 (Feb.).

Forsythe, G. E., and Moler, C. B. (1967). Computer Solution of Linear Algebraic Systems. Englewood Cliffs, N.J.: Prentice-Hall.

Medan, R. T., and Ray, K. S. (1973). Aerodynamic Influence Matrix Program for Aerodynamic Lifting Surface Theory. NASA Report No. TMX 62324.

Nering, E. D. (1963). Linear Algebra and Matrix Theory. New York: John Wiley and Sons.

Ralston, A. (1963). A First Course in Numerical Analysis. New York: McGraw-Hill.

Westlake, J. R. (1968). A Handbook of Numerical Matrix Inversion and Solution of Linear Equations. New York: John Wiley and Sons.

APPENDIX A

A--STANDARDIZED FILE HANDLING PROCEDURES FOR POTFAN PROGRAMS

Standardized FORTRAN procedures and subroutines for opening and closing files have been developed to facilitate using and coding POTFAN programs and the conversion of these codes to different computer systems.

A.1 FILE CREATION

This section describes actions taken before and after any POTFAN program attempts to write a POTFAN file.

Prior to writing any permanent file onto a unit, all POTFAN programs call a system dependent subroutine as follows:

CALL OPENW (NT,IFTYP,ID,IR)

If IR is not zero, then NT and ID are considered subroutine inputs. NT is the logical unit number on which the file will be written and ID is the file creation identifier, which should also be the primary file identification number. If IR is zero, then ID is not considered a subroutine input and NT is only the default unit number. In this case the program reads in ID and NT from a card via 2I5 format. If the value of NT on the card is zero, the subroutine replaces NT with the default value.

If the value of ID determined in either case is then still zero and if it is possible on the computer system being used, the program will replace ID with the current number on the identification number file and also update the identification number file.

In addition to NT, ID, and IR, IFTYP is also input to the program. IFTYP defines the type of file being created according to the following table:

IFTYP	TYPE OF FILE
1	Geometry
2	Boundary condition

3	Influence matrix
4	Velocity matrix
5	Solutions
6	Velocity at force sensing location of N1 segments
7	Velocity at force sensing location of N2 segments
8	Constraint function transformation matrix
9	Zeta plot file
10	Constrained influence matrix
12	Preset solution
15	Inverse or decomposition of influence matrix
16	External velocity
17	Surface pressures
18	Surface velocity

Once ID and NT have been determined, the program opens (if possible on the system being used) the file for writing using a file name determined from ID and IFTYP. On IBM systems, opening a file consists of issuing a DDEF to the operating system. On the INFONET UNIVAC 1108 system, an EQUATE command is involved. This feature eliminates the need for job control cards to handle files on those systems for which FORTRAN programs can open files.

The program then rewinds the file and writes a message indicating which unit has been opened and the value of ID and IFTYP.

After the file has been opened and written upon, it is released by calling another system dependent subroutine as follows:

CALL ENDFIL(NT)

This subroutine writes an end-of-file mark on the unit and (if required by the system being used), releases the unit. The subroutine also writes a message indicating that unit NT has been closed.

A.2 FILE ACCESSING

This section describes actions taken before and after any POTFAN program attempts to read any POTFAN file.

Prior to reading any permanent file from a unit all POTFAN programs call a system dependent subroutine as follows:

CALL OPENR(NT,IFTYP,ID,IR)

If IR is not zero, then NT and ID are considered subroutine inputs. NT is the logical unit number from which the file is read and ID is the file access identifier, which should also be the primary file identification number. If IR is zero, then ID is not considered a subroutine input and NT is only the default unit number. In this case, the program reads in ID and NT from a card via 2I5 format. If the value of NT on the card is zero, the subroutine replaces NT with the default value.

In addition to NT, ID, and IR, IFTYP is also input to the program. IFTYP defines the type of file being read according to the table in the previous section.

Once ID and NT have been determined, the program attempts to open the file using a file name determined from ID and IFTYP. The capability to open a file from a FORTRAN program depends on the system being used. As explained in the previous section, this may involve a DDEF or EQUATE command and can eliminate the need for job control cards to handle files.

The program rewinds the file and writes a message indicating which unit has been opened and the value of ID and IFTYP.

After control is returned to the calling program and the first record of the file has been read, all POTFAN programs check to see if the access identifier is equal to the actual primary file identification number existing on the first record. If not equal, the program writes an informational diagnostic message and proceeds. This feature is meant to be a helpful filekeeping technique for those systems that do not permit automated file control.

After the file has been read and there is no further use for it, it is released by calling another system dependent subroutine as follows:

CALL FILEND(NT)

This subroutine rewinds unit NT and (if required by the system being used) releases the unit.

APPENDIX B

B STANDARDIZED FORMAT OF POTFAN FILES

A standard format has been developed for POTFAN files. This format is applicable to all files except scratch files and plot files. This standard has been developed for the following reasons:

1. to minimize the effects of changes in one POTFAN segment on other POTFAN segments;

2. to allow a program to be developed which can list and/or edit the contents of any POTFAN file; and

3. to promote consistency among POTFAN programs.

Briefly, the standardized POTFAN file consist of one or more records. The first record is called the introductory record and contains miscellaneous data including the primary identification number, a title, and real, integer, and logical parameters reflecting how the data on the remaining records was calculated and/or how it is to be used. The second and subsequent records generally contain the bulk of the data and are called data records. The latter records contain one or more arrays which are always either integer or floating point numbers (i.e. integer and floating point numbers are not mixed on a single record). A detailed description is given below.

First Record (Introductory Record)

This record is created by an unformatted write statement such as the following:

```
WRITE(NT) NCTIME, (CTIME(N),N=1,NCTIME), NTITL,
#(TITL(N),N=1,NTITL),NRECS, (IFORM(N),N=1,NRECS),
#NID, (ID(N),N=1,NID), NLOG, (LOG(N),N=1,NLOG),
#NINT, (INT(N),N=1,NINT), NFLT, (FLT(N),N=1,NFLT)
```

The values of NCTIME, NRECS, NID, NLOG, NINT, and NFLT are all at least one and can vary from file to file even for files of the same type (e.g. NINT may be different on two different geometry files). An explanation of these variables is given below:

NCTIME Number of words in (CTIME)

(CTIME) Creation time in A4 alphanumeric format. Whether or not this array can be filled out depends on the availability of a system dependent subroutine to compute it. This array is used only as a filekeeping aid. It is printed out whenever a file is created or read.

NTITL The number of words in (TITL). Generally NTITL is a multiple of 20.

(TITL) Alphanumeric titling information (e.g. "Delta wing with flaps"). This array is to be written under a format such as (1x,20A4/).

NRECS The number of records (including the first) comprising the file. NRECS is also the number of words in (IFORM).

(IFORM) An integer array indicating the kind of numbers on each record. A value of zero implies an integer and a value of one implies a floating point number. IFORM(1) has no significance.

NID The number of words in (ID)

(ID) Identification number array. ID(NID) is the primary file identification number. In order to keep track of files ID(NID) should be unique for each file. This number is printed out whenever the file is created or read.

NLOG The number of words in (LOG)

(LOG) An array of logical parameters

NINT Number of words in (INT)

(INT) An array of integer parameters

NFLT Number of words in (FLT)

(FLT) An array of floating point parameters. If the remaining data on the file is dependent on Mach number, then FLT(1) is the Mach number.

Second and Subsequent Records (Data Records)

The remaining records of PCTFAN files contain one or more arrays. If the data record contains more than one array, then all arrays on the record must be of the same type (i.e. either integers or real numbers, but not both) and all arrays must have the same number of words. The records also contain array dimensions (J1, J2, and J3) and the total number of words in all arrays on the record (NW). Following are some examples of code used to create data records:

```
      NW = J1*J2*J3
      WRITE(NT) J1,J2,J3,NW,(((A(I,J,K),I=1,J1),J=1,J2),K=1,J3)


      J3 = 2
      NW = J1*J2*J3
      WRITE(NT) J1,J2,J3,NW,((A(I,J),I=1,J1),J=1,J2),
     #((B(I,J),I=1,J1),J=1,J2)


      J2 = 1
      J3 = 1
      NW = J1
      WRITE(NT) J1,J2,J3,NW,(A(I),I=1,NW)


      J2 = 3
      J3 = 1
      NW = 3*J1
      WRITE(NT) J1,J2,J3,NW,(A(I),I=1,J1),(B(I),I=1,J1),
     #(C(I),I=1,J1)
```

Note that in the above examples all dimensions with multiple arrays were written with the leftmost indices varying most rapidly. This practice is always followed unless it is strictly necessary to do otherwise.

No matter how a data record was created, it can be read in by either of the following:

```
      READ(NT) J1,J2,J3,NW,(A(I),I=1,NW)
      READ(NT) J1,J2,J3,NW,(((A(I,J,K),I=1,J1),J=1,J2),K=1,J3)
```

In the former case, the data is packed solidly into core. In the latter case, some a priori knowledge of J1, J2, and

J3 or their maximum allowable values must have been available in order to properly dimension (A). Such a priori knowledge is generally contained as elements of (INT).

Different data records may contain data of different types and may have differing values of J1, J2, J3, and NW.

APPENDIX C

C ARRAY NOTATION

A shorthand notation for referring to arrays in the internal and external documentation of POTFAN programs has been developed. This notation should be made clear by the following examples:

(A)	This implies that A is an array.
(A(N))	This refers to all the words in (A) from 1 through N.
A(N)	This refers only to the Nth word of (A).
(A(I,J))	This refers to all the words in the doubly dimensioned array A for which the first index varies from 1 to I and the second from 1 to J.
A(I,J)	This refers to the element in (A) for which the first index is I and the second is J.
(A(I,J),J=3,K)	This refers to the words of (A) for which the first index is I and the second index varies from 3 to K.
(A(I,*))	This refers to those elements of (A) for which the first index varies from 1 to I and the second index varies from 1 to some value which for some reason cannot be defined.

PROGRAM	CALLED BY	CALLS
ATR	INVRS, ITRM, RIVNS	None
BLKINV	INVRS	MINV
FALL	INVRS, ITRM, MATMLT, MINV	None
HOUSE	INVRS	None
HREC	INVRS	None
INVOUT	INVRS	OPENW, FILEND, TIMEST
INVRS	OPRN, ITRM	OPENW, BLKINV, FALL, HOUSE, HREC, INVOUT, MATPRO, MINV, SOUT, TRIP, SOLVE, ATR
ITRM	OPRN	ATR, FALL, INVRS, MATCOM, MATPRF, MATPRO, OPENW, PART, FILEND, TIMEST, TRIP, OPENR, RELFIL

TABLE 4.1-1. Subprogram Calling Structure

PROGRAM	CALLED BY	CALLS
MATCOM	SCLNIO, ITRM	None
MATMLT	SCLNIO	MATPRC, SOUT, FAIL
MATPRF	ITRM	None
MATPRC	INVRS, ITRM, MATMLT	None
MINV	BLKINV, INVRS	FAIL, TRIP
OPENR	ITRM, RBCS, RCIVNS, RIVNS, RVNS	Machine dependent subroutines only.
OPENW	INVOUT, ITRM, SOUT	Machine dependent subroutines only.
OPRN	SCLNIO	INVRS, ITRM, PART
PART	OPRN, ITRM	None
RBCS	SCLNIO	RELFIL, RHSCOM, OPENR
RCINVS	RVNS	OPENR, RELFIL
RELFIL	ITRM, RBCS, RCIVNS, RIVNS, RVNS	None

TABLE 4.1-1 Subprogram Calling Structure (Cont'd)

PROGRAM	CALLED BY	CALLS
RHSCOM	RBCS	None
RIVNS	SOLNIO	ATB, OPENR, RELFIL
RVNS	SOLNIO	OPENR, RCIVNS, RELFIL, RVNS2
RVNS2	RVNS	None
SOLN	None	SOLNIO, TIMEST
SOLNIO	SOLN	MATCOM, MATMLT, OPRN, RBCS, RIVNS, RVNS
SOLVE	INVRS	ETIME, ETIMEF
SOUT	MATMLT, INVRS	OPENW, FILEND, TIMEST
TIMEST	INVOUT, ITRM, SOUT, SOLN	System dependent subroutines only.
TRIP	MINV, INVRS, ITRM	None

TABLE 4.1-1 Subprogram Calling Structure (Concluded)

COMMON BLOCK NAME	SIZE	USING SUBPROGRAMS
CONSI	10	Nearly all
FSCOM	110	ITRM, RECS, SOUT
INVCOM	70	INVCOT, INVRS, ITRM, MATCOM, MATMLT, OPRN, PART, RBCS, RCIVNS, RHSCOM, RIVNS, RVNS, RVNS2, SOLN, SOLNIO, SOUT
INVFLG	50	ITRM, RCIVNS, RVNS
NIDCOM	864	INVCOT, INVRS, ITRM, MATMLT, RBCS, RCIVNS, RIVNS, RVNS, RVNS2, SOLN, SOLNIO, SOUT
NOCOM	9	INVCOT, INVRS, ITRM, MATCOM, MATMLT, OPRN, PART, RBCS, RCIVNS, RIVNS, RVNS, RVNS2, SOLN, SOLNIO
NTCOM	200	ITRM, RECS, RCIVNS, SOLNIO, SOUT

TABLE 4.3-1 Common Block Usage

FORTRAN VARIABLE	LOGICAL UNIT DESCRIPTION	CURRENT VALUE
INL	Temporary storage device	11
KST	Temporary storage device	13
INCBD	Temporary storage device	14
INLT	Temporary storage device	12
INB	Temporary storage device	15
INBT	Temporary storage device	17
INVR	Temporary storage device or logical unit number for output of inverse or decomposition matrix file	18
NTIR	Temporary storage device	16
NTCP	Printed output device in conversational mode	6
NTP	Printed output device in batch mode	6
NTCR	Conversational mode input device	5
NTR	Batch mode input device	5
RIN	Batch mode input device	5
ROUT	Printed output device in batch mode	6
OUTA( )	Array of logical units for solution output files	User Input
NBCR( )	Array of logical units for constant vector input files	User Input
INPS( )	Array of logical units for preset singularity input files	User Input

TABLE 4.4-1 Summary of Logical Units Used by SOLN Code

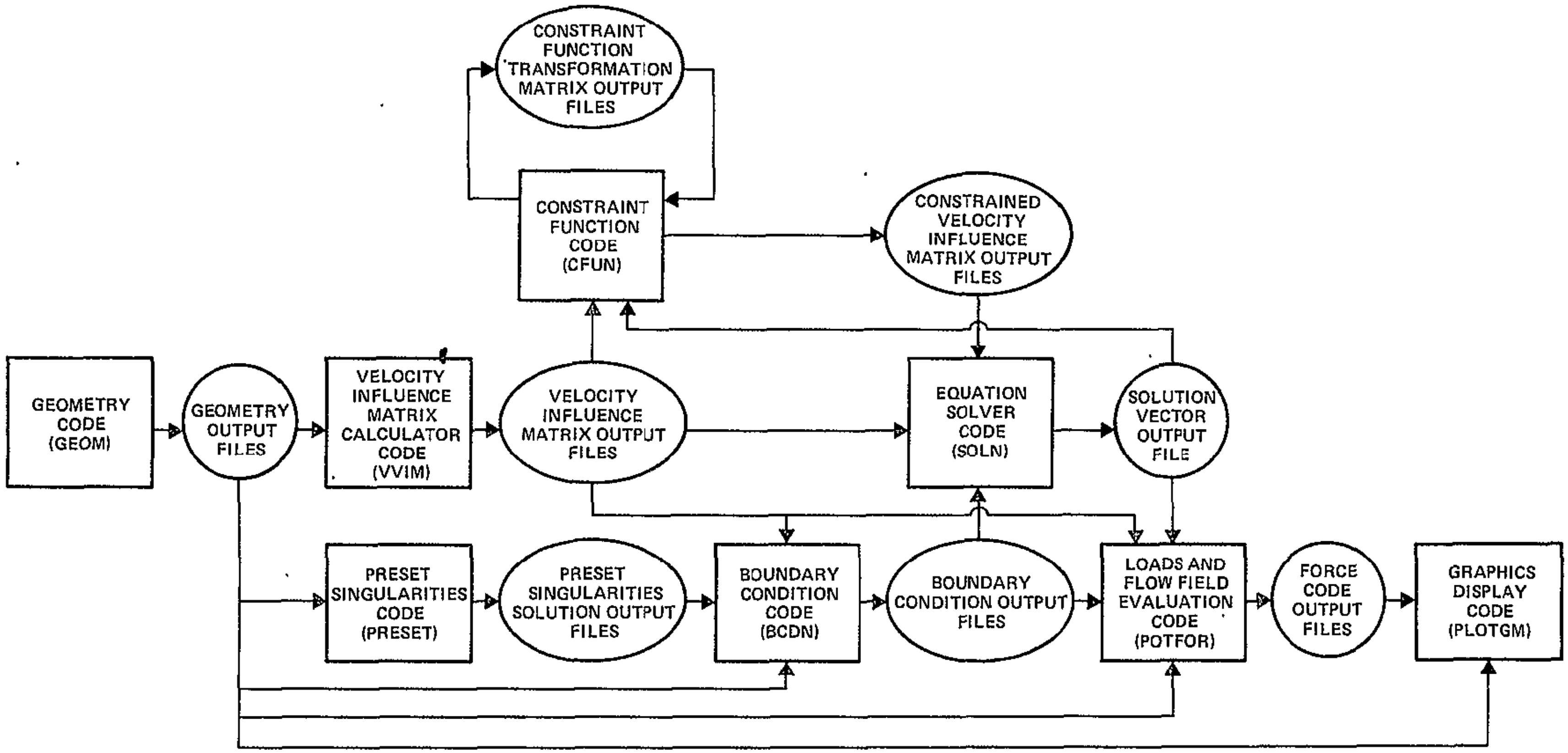

FIGURE 1.0-1 Potential Flow Analysis System Structure

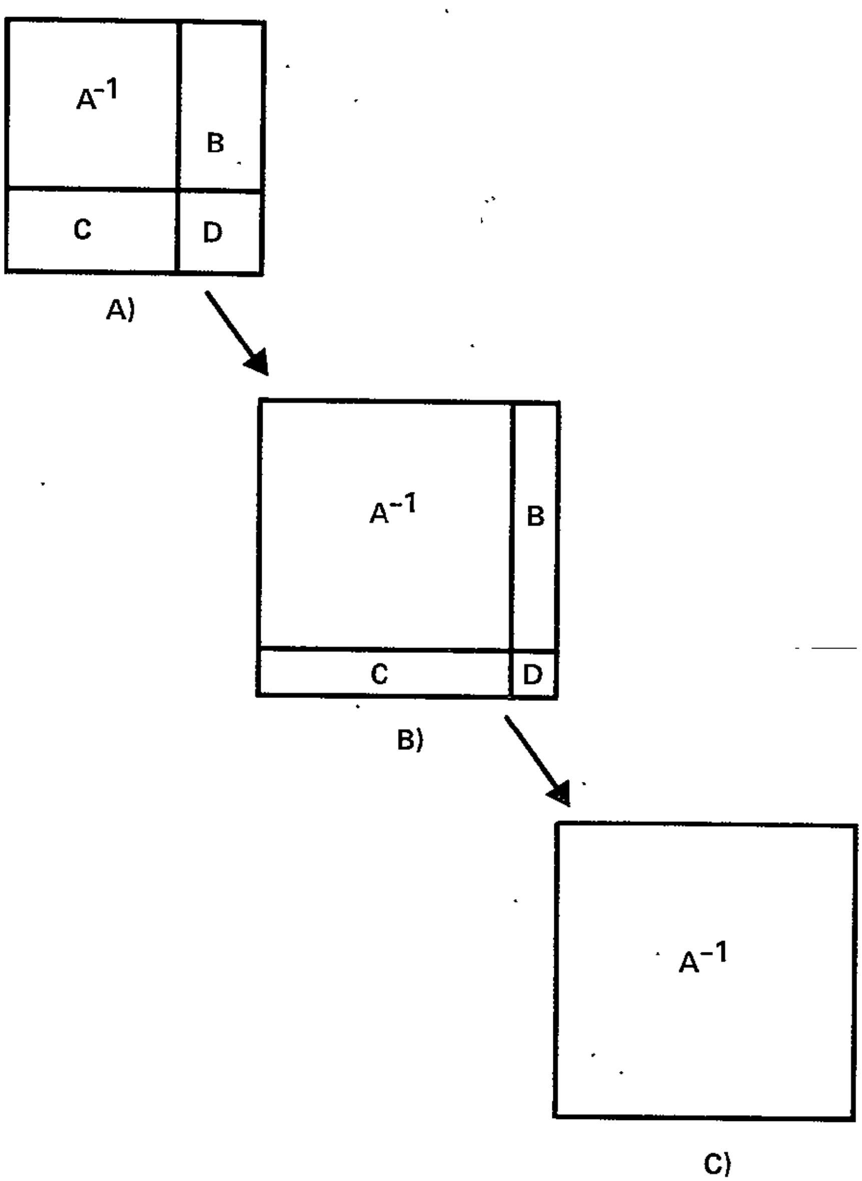

FIGURE 3.1-1 Flow of Partitioning Scheme Solutions

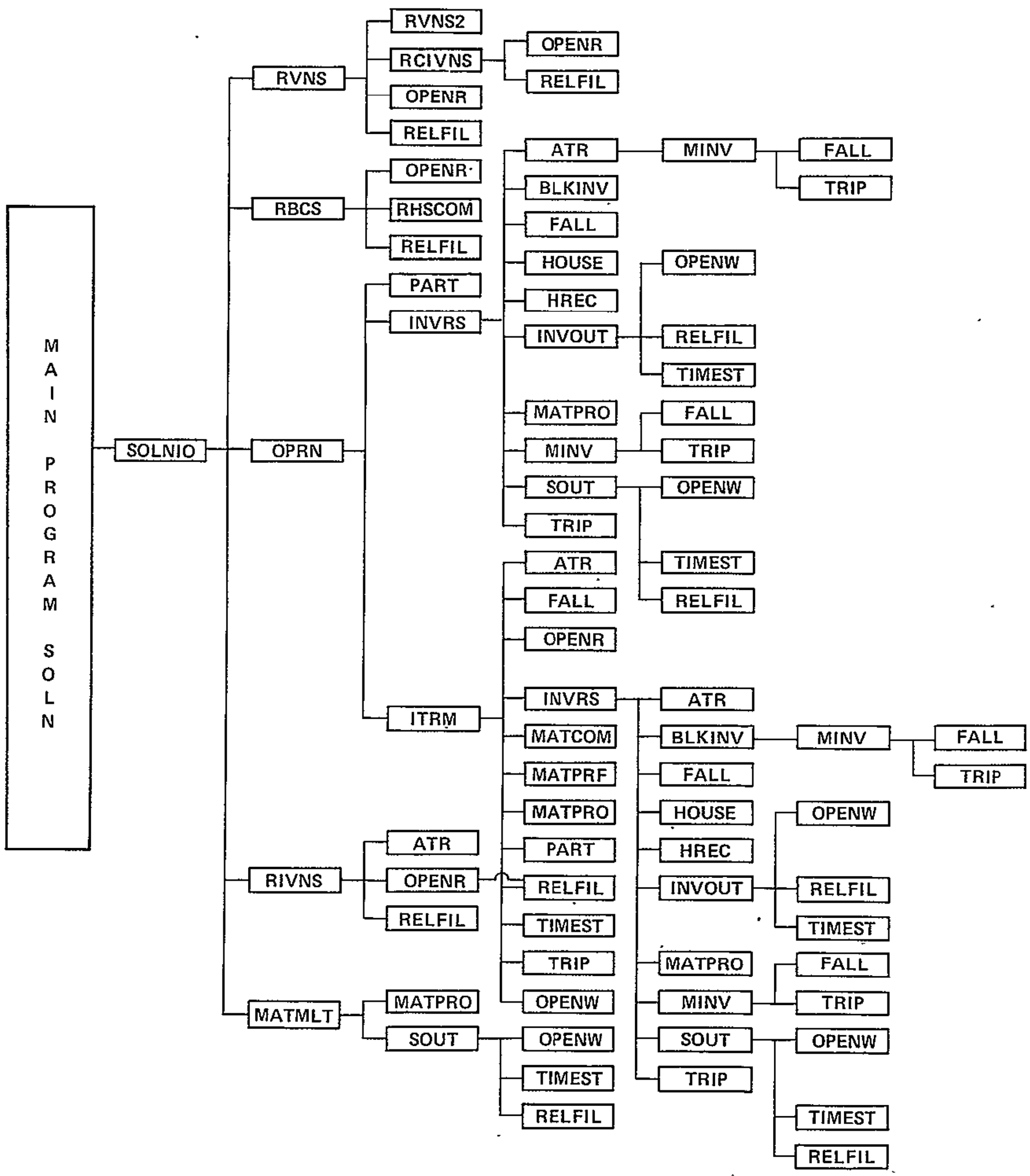

FIGURE 4.1-1 Equation Solver Program Subroutine Structure

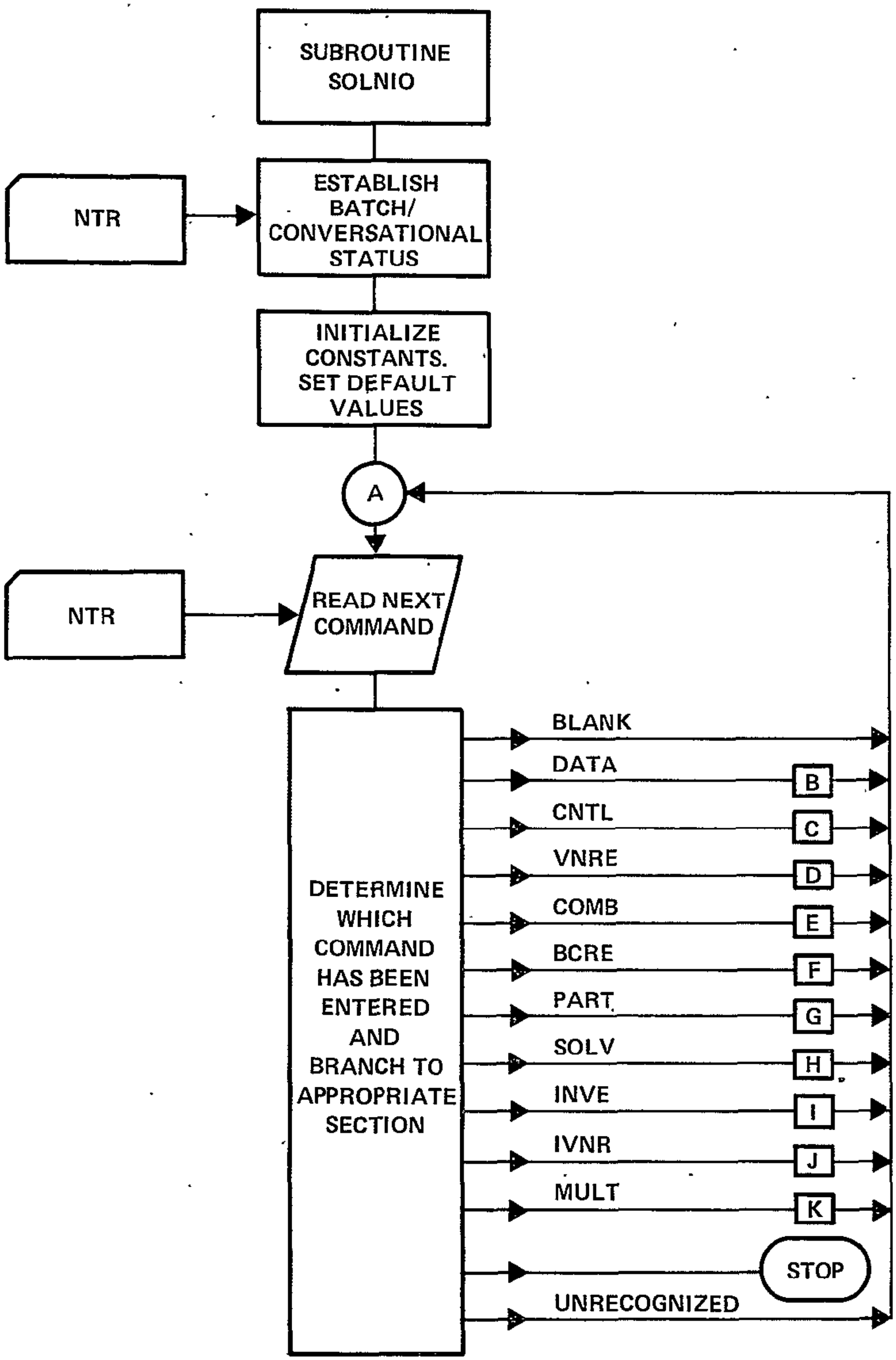

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart

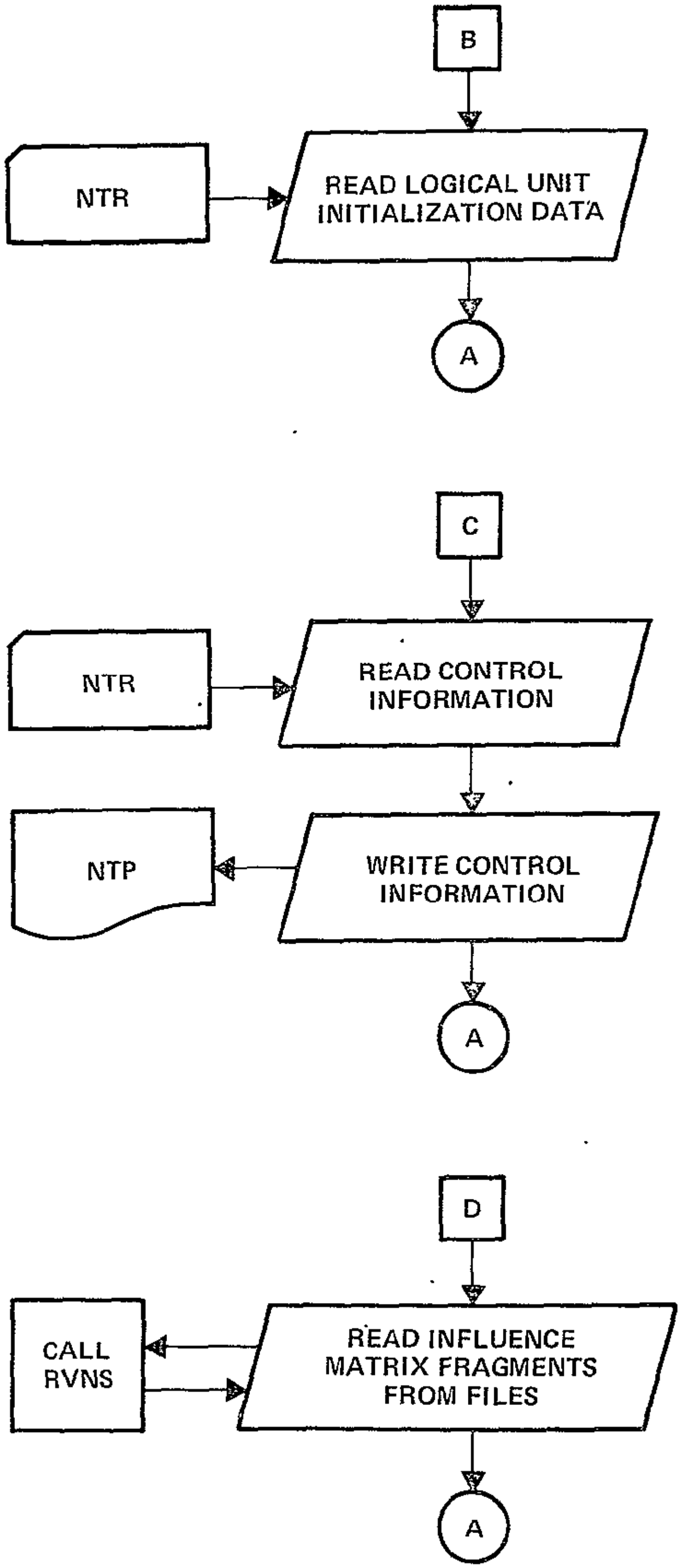

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

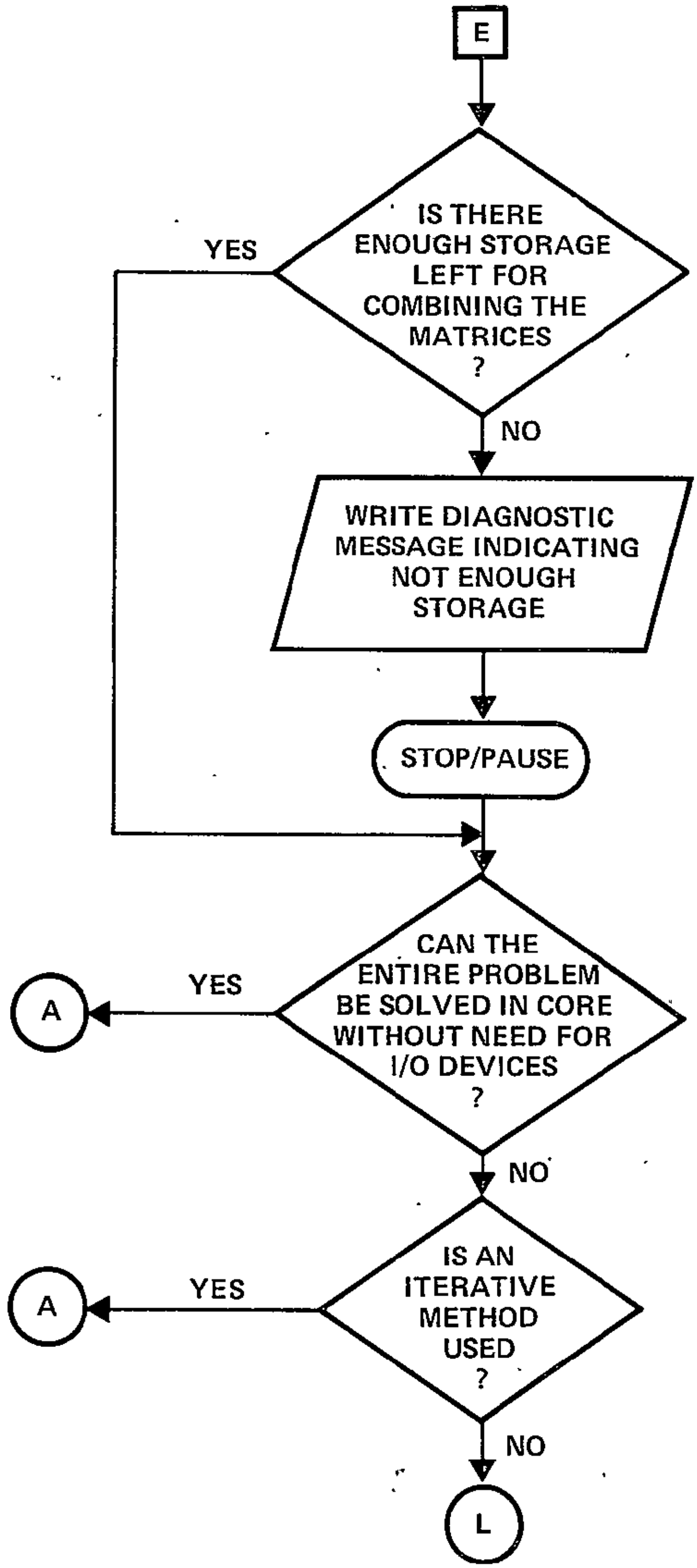

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

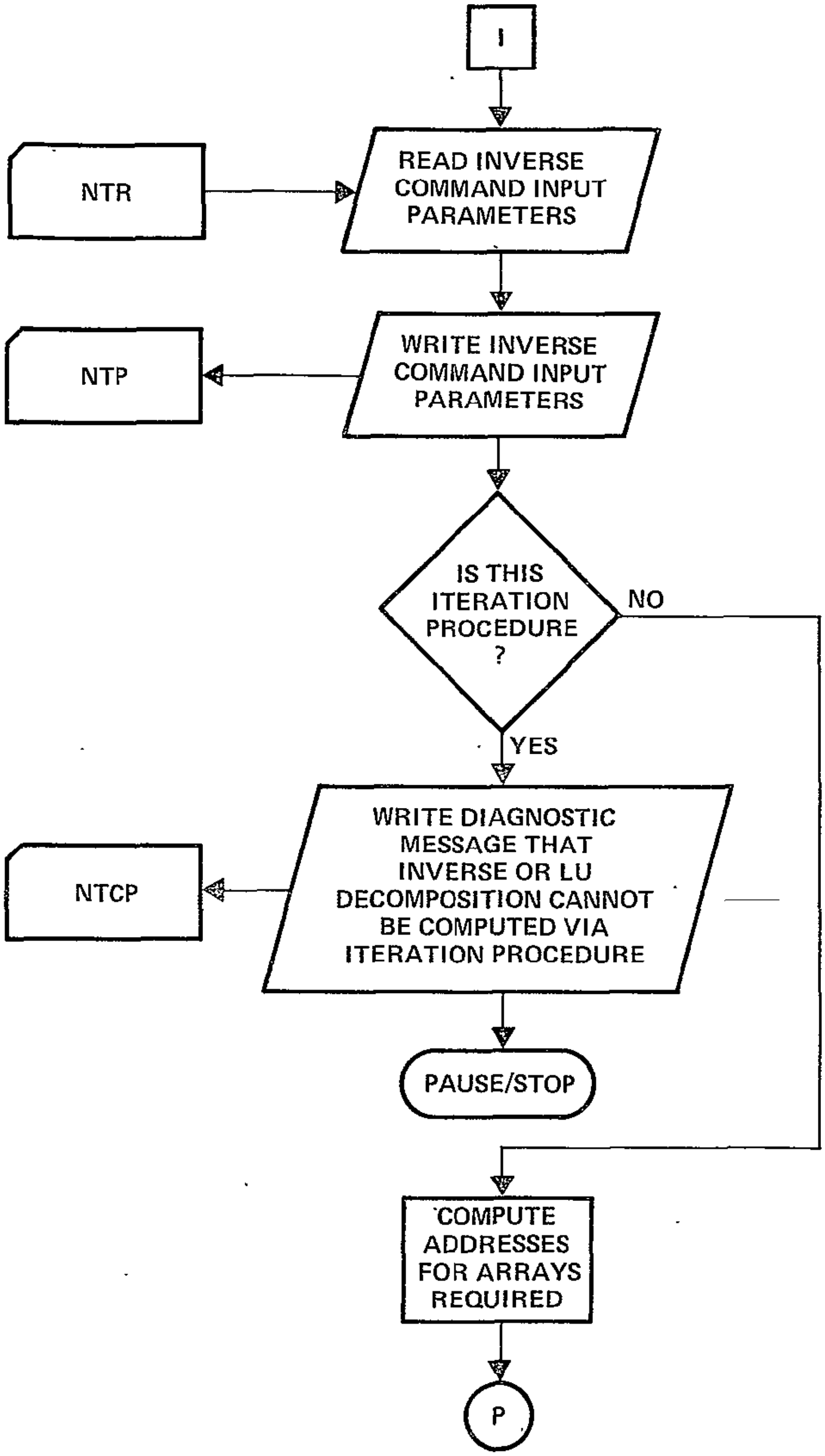

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

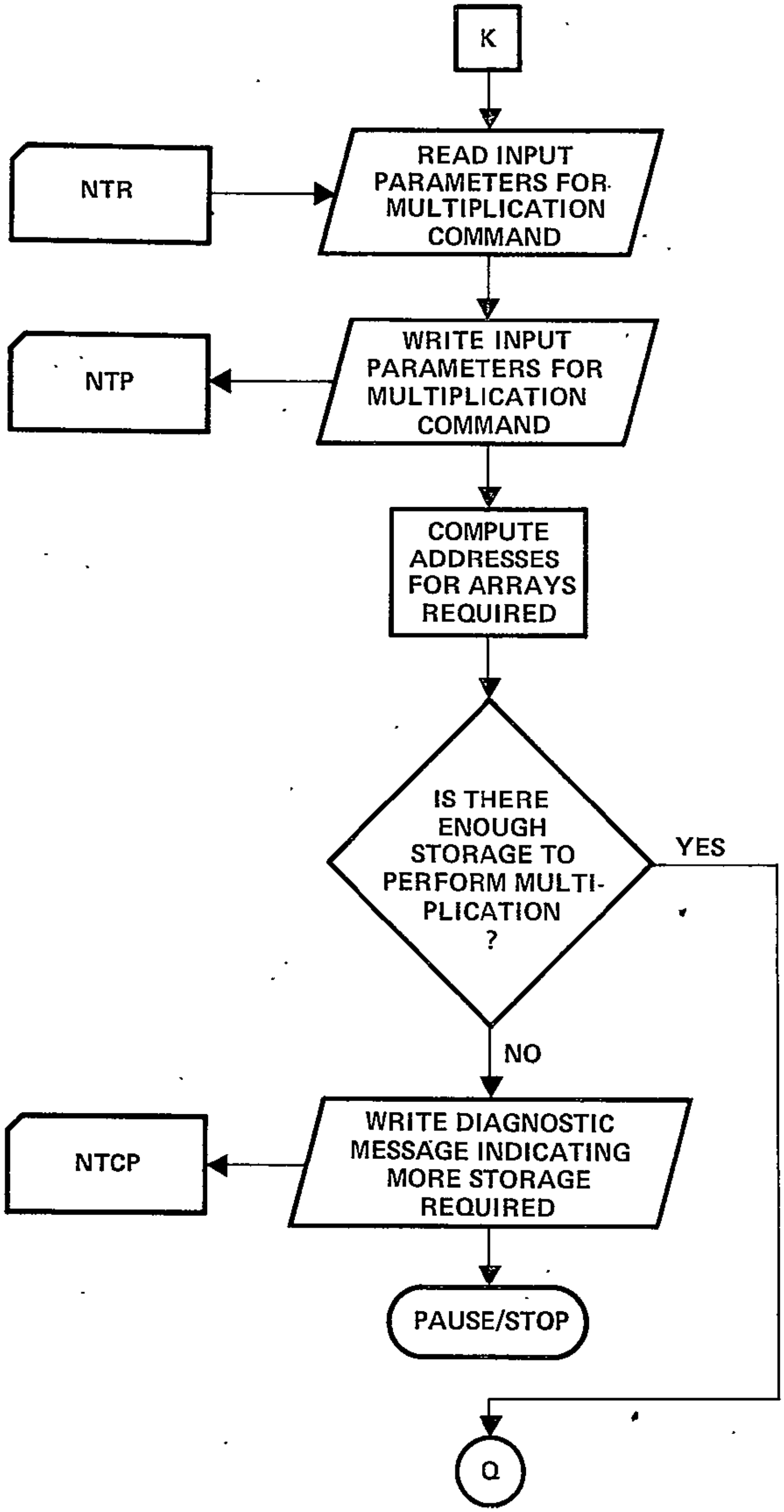

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

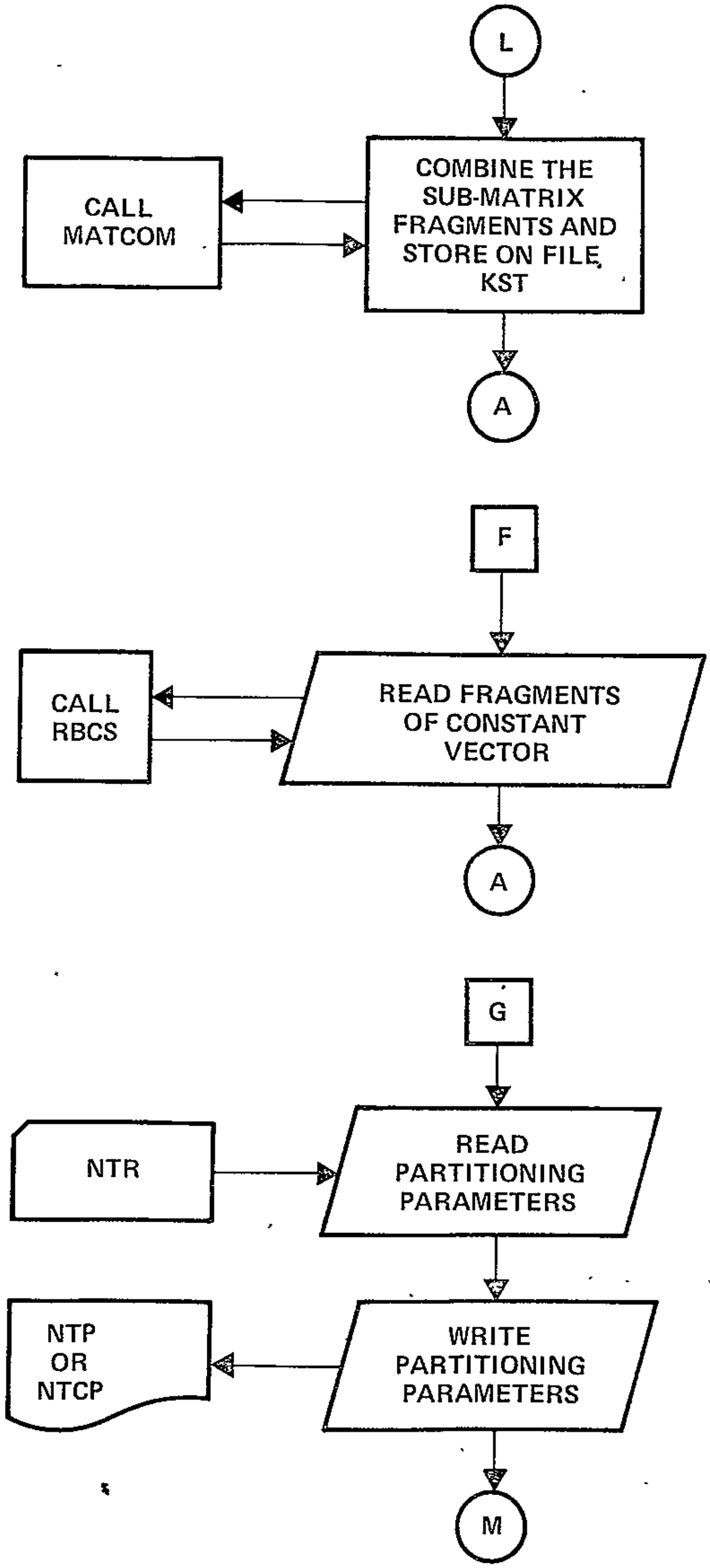

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

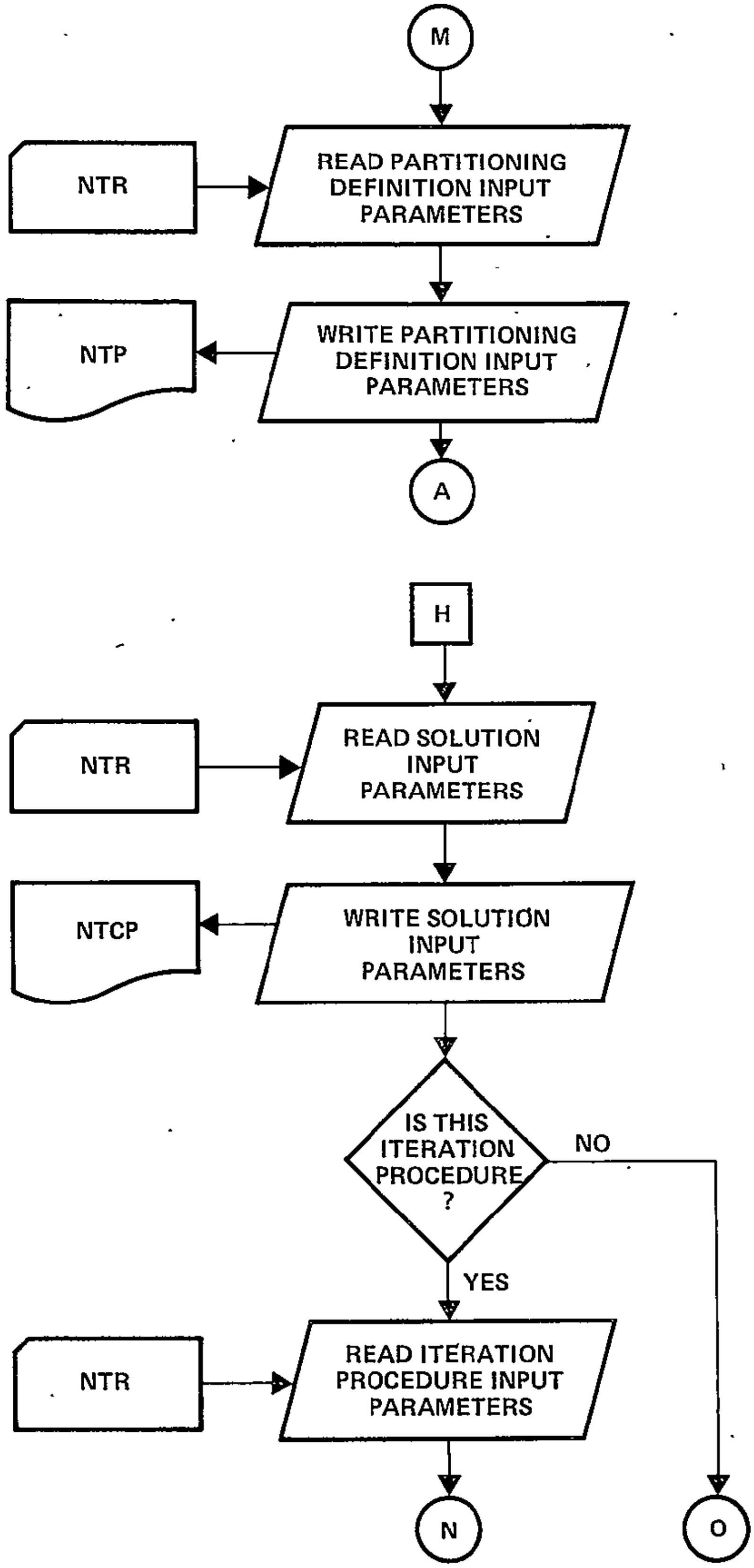

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

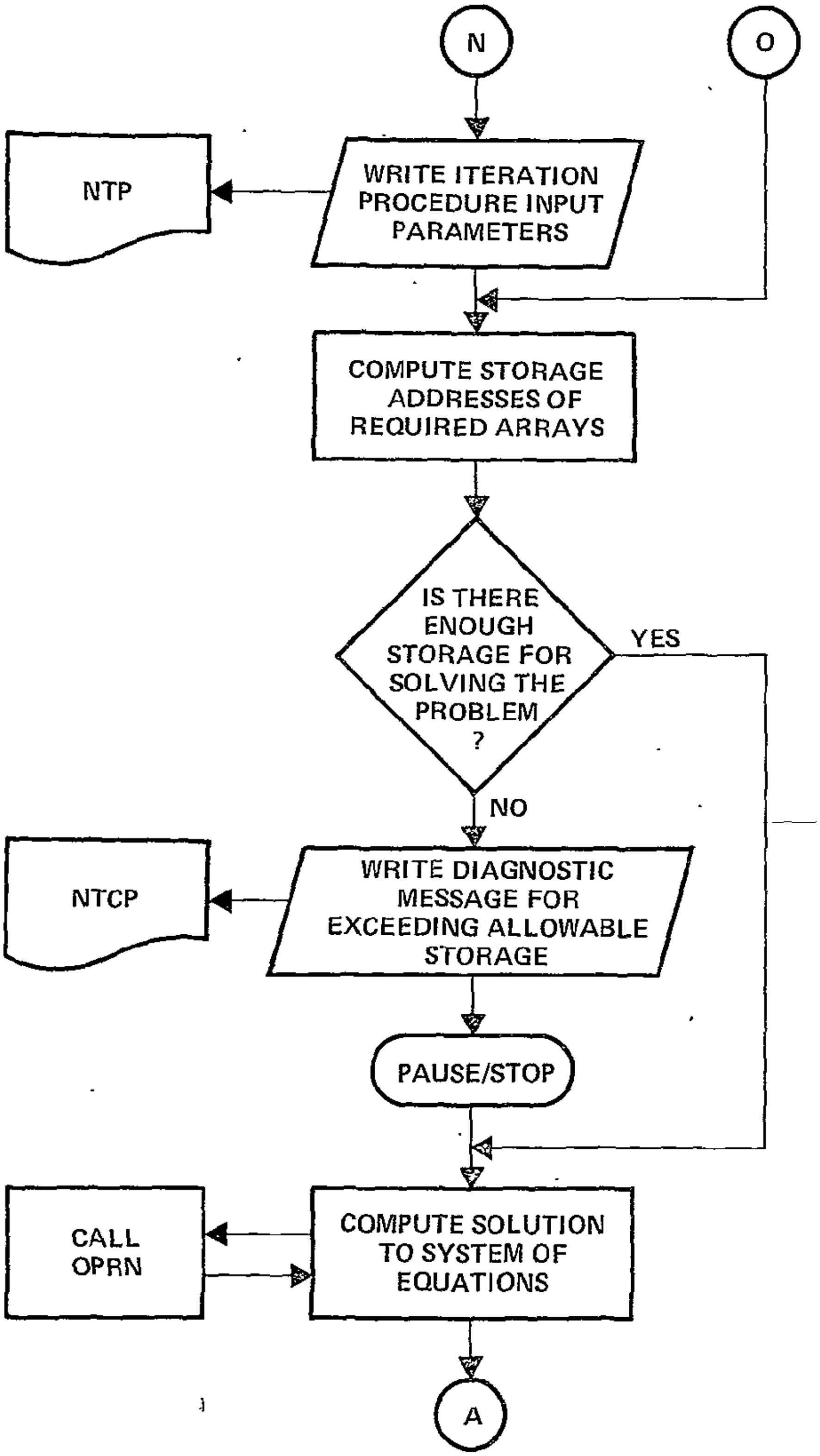

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

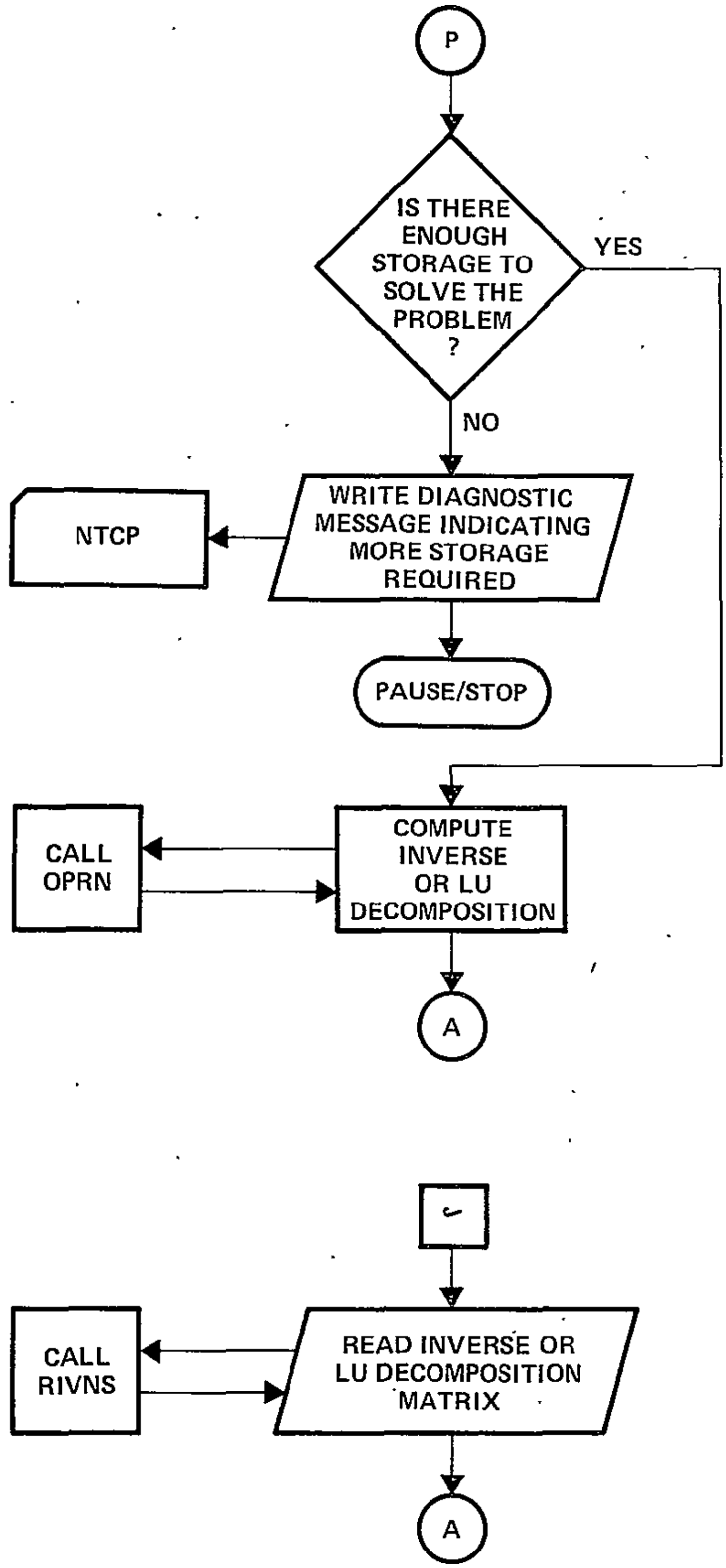

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Cont'd)

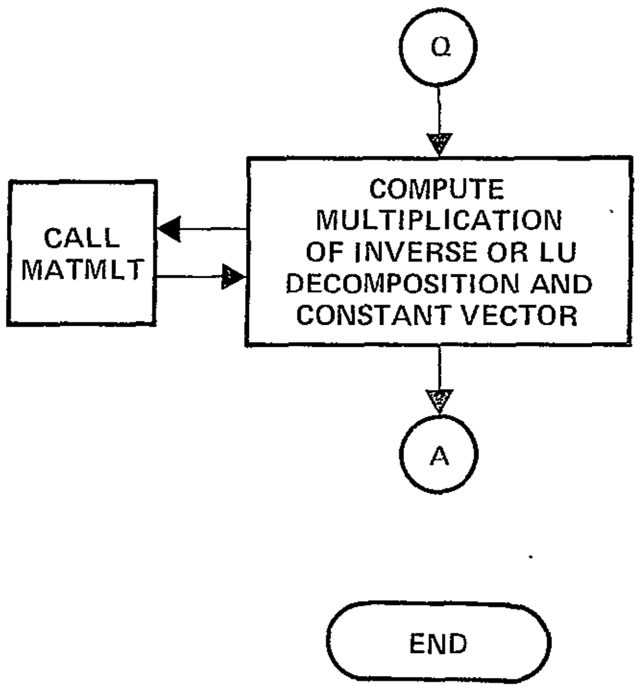

FIGURE 4.2-1 Equation Solver Program Logic Flow Chart (Concluded)

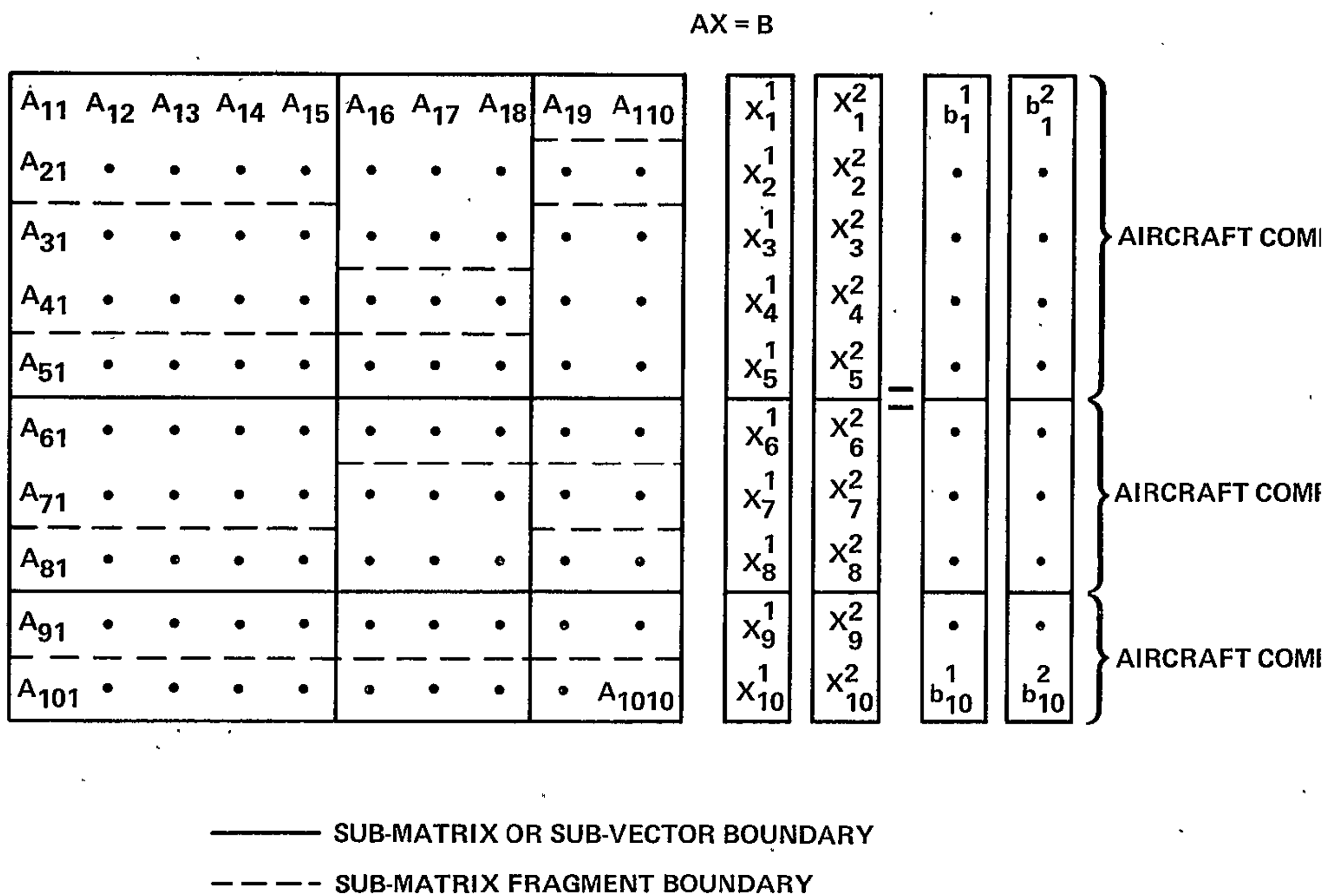

FIGURE 8.0-1 Illustration of the System of Equations Solved for Test Cases 1, 2, 3, and 5; a 10x10 Square Matrix Composed of Three Aircraft Components and 22 Matrix Fragments.

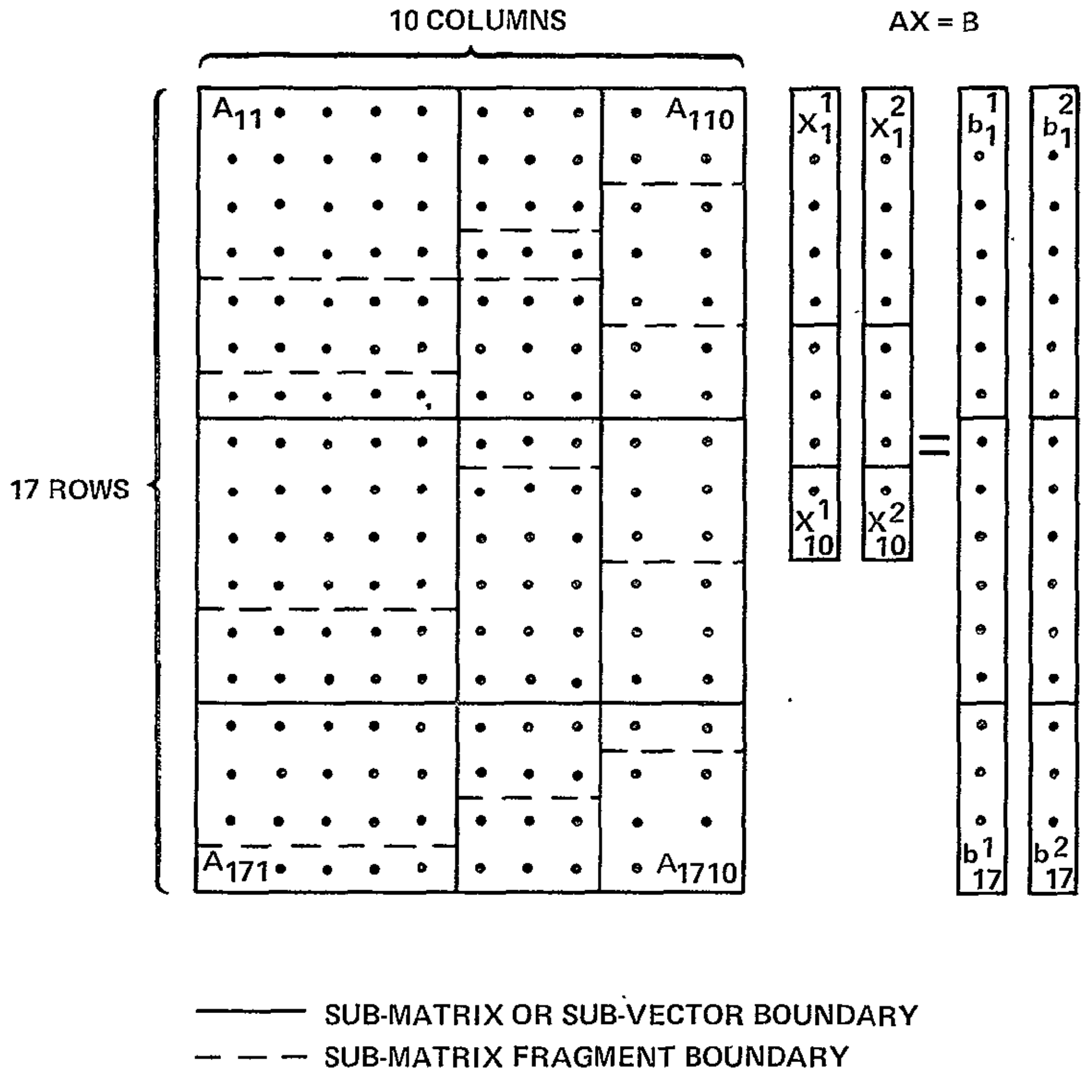

FIGURE 8.0-2. Illustration of the System of Equations Solved for Test Cases 2, 6, and 7; a 17x10 Rectangular Matrix Composed of Three Aircraft Components and 22 Matrix Fragments.

```
      DIMENSION A(150,150),B(150),TITL(5),IFORMV(10),IDV(3),LOGV(12),INT
     1V(12),FLTV(6),IFORMB(9),IDB(2),LOGB(9),INTB(3),FLTB(1),FSC(11)
      DIMENSION NRPC(3),NCPC(3),NFPC(3,3),NRPF(3,3,3),NTVNW(3,3),NTBCW(3
     1)
      LOGICAL LOGV,LOGB
      LOGICAL CONV,BATCH
      COMMON /CONST/ CONV,BATCH,DUMY(4),NTCP,NTP,NTCR,NTR
      DATA TITL/4HTEST,4H CAS,4HE FO,4HR SO,4HLN  /
      DATA IFORMV/0,9*1/,IFORMB/0,8*1/
      DATA LOGV/F,T,8*F,T,F/,INTV/0,5,2,4*0,10,10,1,10,21/,FLTV/0.,1.,0.
     1,0.,0.,0./
      DATA LOGB/4*F,T,F,T,F,F/,INTB/0,10,1/,FLTB/1.0/
      DATA FSC/1.0,10*0.0/
      DATA NCMP/1/,NFPC/1/,MULT/1/,MEXP/3/
      NAMELIST/MDATA/NCMP,NRPC,NCPC,NFPC,NRPF,NTVNW,NTBCW,MULT,MEXP,NSOL
      NTR=5
      NTCR=5
      NTP=6
      NTCP=6
   10 CONTINUE
      READ (5,*,PROMPT=+ENTER N:   +) N
      IF(N.LE.0) STOP 777
      NSQ=N*N
      NSOL=1
      NRPC(1)=N
      NCPC(1)=N
      NRPF(1,1,1)=N
      READ(5,MDATA)
      DO 2 I=1,N
      DO 2 J=1,N
      KK=((N-IABS(I-J))*MULT)**MEXP
      A(I,J)=KK
    2 CONTINUE
      DO 6 I=1,N
      CT=0.
      DO 4 K=1,N
    4 CT=CT+A(I,K)*FLOAT(K)
      B(I)=CT
    6 CONTINUE
```

FIGURE 8.0-3 Program Used to Generate the Matrix and Constant Vectors for Test Cases 1,3,4, and 5.

```
      NTITL=5
      NIDV=3
      NLOGV=12
      NINTV=12
      NFLTV=6
      IDV(1)=1
      IDV(2)=2
      INTV(3)=1
      INTV(12)=21
      NIDB=2
      NLOGB=9
      NINTB=3
      NFLTB=1
      IDB(1)=1
      NDUMI=1
      DUMI=1.
      NVFS=3
      NFSC=11
      N3=1
      NRECSB=1+3*NSOI
      KR=1
      DO 20 IC=1,NCMP
      KC=1
      INTV(2)=NRPC(IC)
      INTV(9)=NRPC(IC)
      LR=KR+NRPC(IC)-1
      DO 50 JC=1,NCMP
      IF(NFPC(JC,IC).EQ.1) NRPF(1,JC,IC)=NRPC(IC)
      LC=KC+NCPC(JC)-1
      NRECSV=1+NFPC(JC,IC)
      NTVN=1
      CALL OPENW(NTVN,3,NTVNW(JC,IC),1)
      IDV(3)=NTVNW(JC,IC)
      INTV(8)=NCPC(JC)
      NFPCIC=NFPC(JC,IC)
      INTV(11)=0
      DO 30 IF=1,NFPCIC
30    IF(NRPF(IF,JC,IC).GT.INTV(11)) INTV(11)=NRPF(IF,JC,IC)
      WRITE(NTVN) NDUMI,NDUMI,NTITL,TITL,NRECSV,(IFORMV(I),I=1,NRECSV),N
     1IDV,IDV,NLOGV,LOGV,NINTV,INTV,NFLTV,FLTV
      WRITE(6,6000) NTITL,TITL,NRECSV,(IFORMV(I),I=1,NRECSV)
6000  FORMAT(2X,'NTVN 1'/2X,I5,5A4/2X,20I5)
```

FIGURE 8.0-3 Program Used to Generate the Matrix and Constant Vectors for Test Cases 1,3,4, and 5 (Cont'd)

```
      WRITE(6,6001) NIDV,IDV,NLOGV,LOGV,NINTV,INTV,NFLTV,FLTV
6001  FORMAT(2X,4I5/2X,I5,12L5/2X,I5,12I5/2X,I5,2X,6E12.5)
      K=KR
      DO 40 IF=1,NFPCIC
      N1=NRPF(IF,JC,IC)
      N2=NCPC(JC)
      L=K+N1-1
      NW=N1*N2
      WRITE(NTVN) N2,N1,N3,NW,((A(I,J),I=K,L),J=KC,LC)
      WRITE(6,6002) N2,N1,N3,NW,((A(I,J),I=K,L),J=KC,LC)
6002  FORMAT(2X,'NTVN 2'/2X,4I5,(/2X,10E12.5))
      K=L+1
40    CONTINUE
      CALL RELFIL(NTVN)
      KC=LC+1
50    CONTINUE
      NTBC=2
      CALL OPENW(NTBC,2,NTBCW(IC),I)
      IDB(2)=NTBCW(IC)
      INTB(2)=NRPC(IC)
      INTB(3)=NSOL
      WRITE(NTBC) NDUMI,NDUMI,NTITL,TITL,NRECSB,(IFORMB(I),I=1,NRECSB),N
     1IDB,IDB,NLOGB,LOGB,NINTB,INTB,NFLTB,FLTB
      WRITE(6,6003) NTITL,TITL,NRECSB,(IFORMB(I),I=1,NRECSB)
6003  FORMAT(2X,'NTBC 1'/2X,I5,5A4/2X,20I5)
      WRITE(6,6004) NIDB,IDB,NLOGB,LOGB,NINTB,INTB,NFLTB,FLTB
6004  FORMAT(2X,3I5/2X,I5,9L5/2X,I5,3I5/2X,I5,2X,11E10.4)
      N1=NRPC(IC)
      DO 100 ISOL=1,NSOL
      DO 200 I=KR,LR
200   B(I)=ISOL*B(I)
      WRITE(NTBC) N1,NDUMI,NDUMI,N1,(B(I),I=KR,LR)
      WRITE(6,6005) N1,NDUMI,NDUMI,N1,(B(I),I=KR,LR)
6005  FORMAT(2X,'NTBC 2'/2X,4I5,(/2X,10E12.5))
      NW=N1*NVFS
      WRITE(NTBC) NFSC,NDUMI,NDUMI,NFSC,FSC
100   CONTINUE
      KR=LR+1
      CALL RELFIL(NTBC)
20    CONTINUE
      STOP
      END
```

FIGURE 8.0-3 Program Used to Generate the Matrix and Constant Vectors for Test Cases 1,3,4, and 5 (Concluded)

```
      DIMENSION A(150,150),B(150),TITL(5),IFORMV(10),IDV(3),LOGV(12),INT
     1V(12),FLTV(6),IFORMB(9),IDB(2),LOGB(9),INTB(3),FLTB(1),FSC(11)
      DIMENSION NRPC(3),NCPC(3),NFPC(3,3),NRPF(3,3,3),NTVNW(3,3),NTBCW(3
     1)
      DIMENSION D(150),VI(6),YC(3),II(6)
      LOGICAL LOGV,LOGB
      LOGICAL CONV,BATCH
      COMMON /CONST/ CONV,BATCH,DUMY(4),NTCP,NTP,NTCR,NTR
      DATA TITL/4HTEST,4H CAS,4HE FO,4HR SO,4HLN  /
      DATA IFORMV/0,9*1/,IFORMB/0,8*1/
      DATA LOGV/F,T,8*F,T,F/,INTV/0,5,2,4*0,10,10,1,10,21/,FLTV/0.,1.,0.,
     1,0.,0.,0./
      DATA LOGB/4*F,T,F,T,F,F/,INTB/0,10,1/,FLTB/1.0/
      DATA FSC/1.0,10*0.0/
      NAMELIST/MDATA/NCMP,NRPC,NCPC,NFPC,NRPF,NTVNW,NTBCW,MULT,MEXP,NSOL
      NAMELIST/SIZE/NR,NC
      NTR=5
      NTCR=5
      NTP=6
      NTCP=6
 10   CONTINUE
      READ(5,SIZE)
      NSQ=NR*NC
      NSOL=1
      READ(5,MDATA)
      JACUM=1
      IACUM=1
      M=0
      DO 3 I=1,NCMP
      M=M+2
      II(M-1)=IACUM
      II(M)=NCPC(I)+IACUM-1
      IACUM=IACUM+NCPC(I)
      VI(M-1)=JACUM
      VI(M)=NRPC(I)+JACUM-1
      JACUM=JACUM+NRPC(I)
 3    CONTINUE
      LMAX=2*NCMP
      M=0
```

FIGURE 8.0-4 Program Used to Generate the Matrix and Constant Vectors for Test Cases 2,6, and 7

```
      DO 1 K=1,3
      K1=K-1
      DO 5 L=1,LMAX
      M=M+1
      D(M)=II(L)**K1
    5 CONTINUE
    1 CONTINUE
      NS=1
      CALL HOUSE(LMAX,3,D,NS,YI,YC)
      DO 2 J=1,NC
      Y=YC(1)+YC(2)*J+YC(3)*J*J
      DO 2 I=1,NR
      A(I,J)=((NR-ABS(I-Y))*MULT)**MEXP
    2 CONTINUE
      DO 6 I=1,NR
      CT=0.
      DO 4 K=1,NC
    4 CT=CT+A(I,K)*FLOAT(K)
      B(I)=CT
    6 CONTINUE
      NTITL=5
      NIDV=3
      NLOGV=12
      NINTV=12
      NFLTV=6
      IDV(1)=1
      IDV(2)=2
      INTV(3)=1
      INTV(12)=21
      NIDB=2
      NLOGB=9
      NINTB=3
      NFLTB=1
      IDB(1)=1
      NDUMI=1
      DUMI=1.
      NVFB=3
      NFBC=11
      N3=1
      NRECBB=1+2*NSOL
      KR=1
      DO 20 IC=1,NCMP
```

FIGURE 8.0-4 Program Used to Generate the Matrix Constant Vectors for Test Cases 2,6, and 7 (Cont'd)

```
      KC=1
      INTV(2)=NRPC(IC)
      INTV(9)=NRPC(IC)
      LR=KR+NRPC(IC)-1
      DO 50 JC=1,NCMP
      IF(NFPC(JC,IC).EQ.1) NRPF(1,JC,IC)=NRPC(IC)
      LC=KC+NCPC(JC)-1
      NRECSV=1+NFPC(JC,IC)
      NTVN=1
      CALL OPENW(NTVN,3,NTVNW(JC,IC),1)
      IDV(3)=NTVNW(JC,IC)
      INTV(8)=NCPC(JC)
      NFPCIC=NFPC(JC,IC)
      INTV(11)=0
      DO 30 IF=1,NFPCIC
30    IF(NRPF(IF,JC,IC).GT.INTV(11)) INTV(11)=NRPF(IF,JC,IC)
      WRITE(NTVN) NDUMI,NDUMI,NTITL,TITL,NRECSV,(IFORMV(I),I=1,NRECSV),N
     1IDV,IDV,NLOGV,LOGV,NINTV,INTV,NFLTV,FLTV
      WRITE(6,6000) NTITL,TITL,NRECSV,(IFORMV(I),I=1,NRECSV)
6000  FORMAT(2X,'NTVN 1'/2X,I5,5A4/2X,20I5)
      WRITE(6,6001) NIDV,IDV,NLOGV,LOGV,NINTV,INTV,NFLTV,FLTV
6001  FORMAT(2X,4I5/2X,I5,12L5/2X,I5,12I5/2X,I5,2X,6F12.5)
      K=KR
      DO 40 IF=1,NFPCIC
      N1=NRPF(IF,JC,IC)
      N2=NCPC(JC)
      L=K+N1-1
      NW=N1*N2
      WRITE(NTVN) N2,N1,N3,NW,((A(I,J),I=K,L),J=KC,LC)
      WRITE(6,6002) N2,N1,N3,NW,((A(I,J),I=K,L),J=KC,LC)
6002  FORMAT(2X,'NTVN 2'/2X,4I5,(/2X,10E12.5))
      K=L+1
40    CONTINUE
      CALL RELFIL(NTVN)
      KC=LC+1
50    CONTINUE
      NTBC=2
      CALL OPENW(NTBC,2,NTBCW(IC),1)
      IDB(2)=NTBCW(IC)
      INTB(2)=NRPC(IC)
      INTB(3)=NSOL
      WRITE(NTBC) NDUMI,NDUMI,NTITL,TITL,NRECSB,(IFORMB(I),I=1,NRECSB),N
```

FIGURE 8.0-4 Program Used to Generate the Matrix and Constant Vectors for Test Cases 2,6, and 7 (Cont'd)

```
     1IDB,IDB,NLOGB,LOGB,NINTB,INTB,NFLTB,FLTB
      WRITE(6,6003) NTITL,TITL,NRECSB,(IFORMB(I),I=1,NRECSB)
 6003 FORMAT(2X,'NTBC 1'/2X,I5,5A4/2X,20I5)
      WRITE(6,6004) NIDB,IDB,NLOGB,LOGB,NINTB,INTB,NFLTB,FLTB
 6004 FORMAT(2X,3I5/2X,I5,9L5/2X,I5,3I5/2X,I5,2X,11E10.4)
      N1=NRPC(IC)
      DO 100 ISOL=1,NSOL
      DO 200 I=KR,LR
  200 B(I)=ISOL*B(I)
      WRITE(NTBC) N1,NDUMI,NDUMI,N1,(B(I),I=KR,LR)
      WRITE(6,6005) N1,NDUMI,NDUMI,N1,(B(I),I=KR,LR)
 6005 FORMAT(2X,'NTBC 2'/2X,4I5,(/2X,10E12.5))
      NW=N1*NVFS
      WRITE(NTBC) NFSC,NDUMI,NDUMI,NFSC,FSC
  100 CONTINUE
      KR=LR+1
      CALL RELFIL(NTBC)
   20 CONTINUE
      STOP
      END
```

FIGURE 8.0-4 Program Used to Generate the Matrix and Constant Vectors for Test Cases 2,6, and 7 (Concluded)

```
 10
$MDATA NCMP=3,NRPC=5,3,2,NCPC=5,3,2,NFPC=3,3,3,2,2,3,2,2,2,
NRFF=2,2,1,3,1,1,1,1,3,2,1,0,1,2,0,1,1,1,1,1,0,1,1,0,1,1,0,
NTVNW=7059,7060,7061,7062,7063,7064,7065,7066,7067,
NTRCW=7068,7069,7070,MULT=1,MEXP=7,NSOL=2 $END
```

FIGURE 8.0-5 Input Used to Run SIEST Program

```
$SIZE NR=17,NC=10 $END
$MDATA NCMP=3,NRPC=7,6,4,NCPC=5,3,2,NFPC=3,3,3,2,2,3,2,2,2,
NRPF=4,2,1,3,1,3,2,3,2,4,2,0,1,5,0,3,2,1,3,1,0,2,2,0,1,3,0,
NTVNW=7077,7078,7079,7080,7081,7082,7083,7084,7085,
NTBCW=7086,7087,7088,MULT=1,MEXP=7,NSOL=2 $END
```

FIGURE 8.0-6 Input Used to Run STESTR Program

```
TRUE
     SAMPLE CASE 1 FOR THE SOLN CODE.
     SOLUTION BY DIRECT
     LU DECOMPOSITION METHOD.
     BATCH MODE.
     THREE AIRCRAFT COMPONENTS.
     TWO CONSTANT VECTORS.
CNTL
     3
     READ INFLUENCE SUB-MATRIX FILES.
VNRF
 7059
 7060
 7061
 7062
 7063
 7064
 7065
 7066
 7067
     SINCE THE ENTIRE MATRIX FITS IN
     CORE, COMB COMMAND IS NOT
     NEEDED.
     READ CONSTANT SUB-VECTOR FILES.
BCRE
 7068
 7069
 7070
     SOLVE BY LU DECOMPOSITION,
     AND DEFINE THE SOLUTION
     SUB-VECTOR OUTPUT FILES.
SOLV
     0
 7071
 7072
 7073
0
     ALL DONE.
STOP
```

FIGURE 8.1-1 Input for SCLN Code for Test Case No. 1

```
POTFAN EQUATION SOLVING PROGRAM (SOLN),  VERSION 1.1
*****************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76     11:05:42

+!       SAMPLE CASE 1 FOR THE SOLN CODE.
+!       SOLUTION BY DIRECT
+!       LU DECOMPOSITION METHOD.
+!       BATCH MODE.
+!       THREE AIRCRAFT COMPONENTS.
+!       TWO CONSTANT VECTORS.
+!  CNTL
NCMP=    3   ITERAT=     0
+!       READ INFLUENCE SUB-MATRIX FILES.
+!  VNRE
FILE 7059,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7060,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7061,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7062,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7063,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7064,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7065,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7066,VN=PNC                HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
```

FIGURE 8.1-2 Output for SOLN Code for Test Case No. 1

```
UNIT   1 REWOUND AND RELEASED
FILE 7067,VH-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:08:54
UNIT   1 REWOUND AND RELEASED
+:        SINCE THE ENTIRE MATRIX FITS IN
+:      . CORE, COMB COMMAND IS NOT
+:        NEEDED.
+:        READ CONSTANT SUB-VECTOR FILES.
+:   BCRE
FILE 7068,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT   2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
FILE 7069,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT   2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
FILE 7070,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT   2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
+:        SOLVE BY LU DECOMPOSITION,
+:        AND DEFINE THE SOLUTION
+:        SUB-VECTOR OUTPUT FILES.
+:   SOLV
METHOD=
IPS=.    0
IPRINT=
(OUTA,OUTANI,I=1,NCMP)=
 7071
 7072
 7073
ENTER SUB. PART    #FRAGS=  22      INV= -2
MATRIX LOADED FROM UNIT  13  USING   22  READS AND  176   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

ENTER SUB. TRIP.   N=   10
EXIT TRIP

OUTPUT
SOLUTION TIME (SEC.)       .019


SOLUTION VECTOR NO.  1
```

FIGURE 8.1-2 Output for SCLN Code for Test Case No. 1 (Cont'd)

```
----------------------

FILE 7071,SO-PNC/LIB$        HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:06:23

UNIT NO,  3
COMPONENT NO,   1  SOLUTIONS:
        .1000000E+01   .2000000E+01   .3000000E+01   .4000000F+01   .5000000E+0
UNIT  3 ENDFILED AND RELEASED
FILE 7072,SO-PNC/LIB$        HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:06:24

UNIT NO,  3
COMPONENT NO,   2  SOLUTIONS:
        .6000000E+01   .7000000F+01   .8000000E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7073,SO-PNC/LIB$        HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:06:24

UNIT NO,  3
COMPONENT NO,   3  SOLUTIONS:
        .9000001E+01   .1000000E+02
UNIT  3 ENDFILED AND RELEASED

OUTPUT
SOLUTION TIME (SEC.)         .003


SOLUTION VECTOR NO.  2
----------------------

FILE 7071,SO-PNC/LIB$        HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO,  3
COMPONENT NO,   1  SOLUTIONS:
        .2000000E+01   .4000000E+01   .6000000E+01   .8000001E+01   .9999999E+0
UNIT  3 ENDFILED AND RELEASED
FILE 7072,SO-PNC/LIB$        HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO,  3
COMPONENT NO,   2  SOLUTIONS:
        .1200000E+02   .1400000F+02   .1600000E+02
```

FIGURE 8.1-2 Output for SCLN Code for Test Case No. 1 (Cont'd)

```
UNIT   3 ENDFILED AND RELEASED
FILE 70/3,SO-PNC/LIB*      HAS BEEN OPENED FOR WRITING ON UNIT   3

UNIT NO.   3
COMPONENT NO.    3   SOLUTIONS:
        .1000000E+02    .2000000E+02
UNIT   3 ENDFILED AND RELEASED
+:      ALL DONE.
+:   STOP
 STOP 777
```

FIGURE 8.1-2 Output for SCLN Code for Test Case No. 1 (Concluded)

```
TRUE
     SAMPLE CASE 2 FOR THE SOLN CODE.
     RECTANGULAR MATRIX (17X10)
     SOLUTION BY DIRECT
     HOUSEHOLDER METHOD.
     BATCH MODE.
     THREE AIRCRAFT COMPONENTS.
CNTL
     3
     READ INFLUENCE SUB.MATRIX FILES.
VNRE
 7077
 7078
 7079
 7080
 7081
 7082
 7083
 7084
 7085
     ASSUME THAT THE MATRIX WILL
     NOT FIT IN CORE ALL AT ONCE.
COMB
     READ CONSTANT SUB-VECTOR FILES.
BCRE
 7086
 7087
 7088
     SOLVE BY HOUSEHOLDER METHOD,
     AND DEFINE THE SOLUTION
     SUB-VECTOR OUTPUT FILES.
SOLV
     1
 7074
 7075
 7076
0
     ALL DONE.
STOP
```

FIGURE 8.2-1 Input for SOLN Code for Test Case No. 2

```
POTFAN EQUATION SOLVING PROGRAM (SOLN),  VERSION 1.1
************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76    11:07:21

+:      SAMPLE CASE 2 FOR THE SOLN CODE.
+:      RECTANGULAR MATRIX (17X10)
+:      SOLUTION BY DIRECT
+:      HOUSEHOLDER METHOD.
+:      BATCH MODE.
+:      THREE AIRCRAFT COMPONENTS.
+:  CNIL
NCMP=     3  ITERAT=    0
+:      READ INFLUENCE SUB-MATRIX FILES.
+:  VNRI
FILE 7077,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7078,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7079,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7080,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7081,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7082,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7083,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7084,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
```

FIGURE 8.2-2 Output for SOLN Code for Test Case No. 2

```
UNIT  1 REWOUND AND RELEASED
FILE 7085,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
+1       ASSUME THAT THE MATRIX WILL
+1       NOT FIT IN CORE ALL AT ONCE.
+1   COMB
+1       READ CONSTANT SUB-VECTOR FILES.
+1   RCRE
FILE 7086,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7087,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7088,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:09:01
UNIT  2 REWOUND AND RELEASED
+1       SOLVE BY HOUSEHOLDER METHOD,
+1       AND DEFINE THE SOLUTION
+1       SUB-VECTOR OUTPUT FILES.
+1   SOLV
METHOD=     1
IPS=     0
IPRINT=     0
(OUTA,OUTANT,I=1,NCMP)=
 7074     0
 7075     0
 7076     0
ENTER SUB. PART   #FRAGS=  22     INV= =1
MATRIX LOADED FROM UNIT  13  USING   22  READS AND  176   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

ENTER SUB. HOUSE.    17 ROWS    10 COLUMNS    2 SOLUTIONS
EXIT HOUSE

OUTPUT
SOLUTION TIME (SEC.)       .034

***  SOLUTION TIME IS THAT FOR ALL SOLUTIONS
```

FIGURE 8.2-2 Output for SOLN Code for Test Case No. 2 (Cont'd)

```
SOLUTION VECTOR NO. 1
---------------------

FILE 7074,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:07:46

UNIT NO.  3
COMPONENT NO.   1  SOLUTIONS:
       .1000000E+01   .2000000E+01   .3000000E+01   .4000001E+01   .4999999E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7075,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:07:47

UNIT NO.  3
COMPONENT NO.   2  SOLUTIONS:
       .6000000E+01   .7000000E+01   .8000000E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7076,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:07:48

UNIT NO.  3
COMPONENT NO.   3  SOLUTIONS:
       .9000000E+01   .1000000E+02
UNIT  3 ENDFILED AND RELEASED

SOLUTION VECTOR NO. 2
---------------------

FILE 7074,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO.  3
COMPONENT NO.   1  SOLUTIONS:
       .2000001E+01   .4000000E+01   .6000000E+01   .8000001E+01   .9999999E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7075,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO.  3
COMPONENT NO.   2  SOLUTIONS:
       .1200000E+02   .1400000E+02   .1600000E+02
UNIT  3 ENDFILED AND RELEASED
FILE 7076,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
```

FIGURE 8.2-2 Output for SCLN Code for Test Case No. 2 (Cont'd)

```
UNIT NO.  3
COMPONENT NO.   3  SOLUTIONS:
       .1400000E+02   .2000000E+02
UNIT  3 ENDFILED AND RELEASED
+:      ALL DONE.
+:  STOP
 STOP 777
```

FIGURE 8.2-2 Output for SCLN Code for Test Case No. 2 (Concluded)

```
TRUE
     SAMPLE CASE 3 FOR SOLN CODE,
     SOLUTION BY CREATING MATRIX
     DECOMPOSITION FILE, READING IT
     BACK IN, THEN PERFORMING FORWARD
     ELIMINATION AND BACKWARD
     SUBSTITUTION BY MULT COMMAND.
     BATCH MODE.
     THREE AIRCRAFT COMPONENTS.
CNTI
     3
     READ INFLUENCE MATRIX FILES.
VNRE
 7059
 7060
 7061
 7062
 7063
 7064
 7065
 7066
 7067
     SINCE ENTIRE MATRIX FITS IN CORE,
     COMB COMMAND IS NOT NEEDED.
     PERFORM MATRIX DECOMPOSITION, AND
     DEFINE THE DECOMPOSITION OUTPUT
     FILE AND ITS TITLE.
INVE
     0 7095     1
    10
     DECOMPOSITION MATRIX FOR SAMPLE CASE 3.
     USER COULD STOP HERE AND PERFORM
     REMAINING COMMANDS LATER, OR HE
     MAY CONTINUE.
     READ CONSTANT SUB-VECTOR FILES.
BCRE
 7068
 7069
 7070
     READ DECOMPOSITION MATRIX FILE
     BACK IN.
IVNR
 7095
     PERFORM FORWARD ELIMINATION AND
     BACKWARD SUBSTITUTION, AND DEFINE
```

FIGURE 8.3-1 Input for SOLN Code for Test Case No. 3

```
    SOLUTION SUB-VECTOR OUTPUT FILES.
MULT
 7077
 7078
 7079
    ALL DONE.
STOP
```

FIGURE 8.3-1 Input for SOLN Code for Test Case No. 3 (Concluded)

```
POTFAN EQUATION SOLVING PROGRAM (SOLN), VERSION 1.1
****************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76    11:08:05

+:       SAMPLE CASE 3 FOR SOLN CODE.
+:       SOLUTION BY CREATING MATRIX
+:       DECOMPOSITION FILE, READING IT
+:       BACK IN, THEN PERFORMING FORWARD
+:       ELIMINATION AND BACKWARD
+:       SUBSTITUTION BY MULT COMMAND.
+:       BATCH MODE.
+:       THREE AIRCRAFT COMPONENTS.
+:  CNTL
NCMP=     3  ITERAT=     0
+:       READ INFLUENCE MATRIX FILES.
+:  VNRE
FILE 7059,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7060,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7061,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7062,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7063,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7064,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7065,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
```

FIGURE 8.3-2 Output for SOLN Code for Test Case No. 3

```
FILE 7066,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7067,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76     04:08:54
UNIT  1 REWOUND AND RELEASED
+:      SINCE ENTIRE MATRIX FITS IN CORE,
+:      COMB COMMAND IS NOT NEEDED.
+:      PERFORM MATRIX DECOMPOSITION, AND
+:      DEFINE THE DECOMPOSITION OUTPUT
+:      FILE AND ITS TITLE.
+:  INVE
METHOD=     0  NTINVW= 7095  IDC=    1
ENTER SUB. PART    #FRAGS=  22     INV= =2
MATRIX LOADED FROM UNIT  13  USING   22  READS AND  176   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

***  MATRIX INVERSE DEMANDED

ENTER SUB. TRIP.  N=   10
EXIT TRIP
          INVERSION OR DECOMPOSITION TIME (SEC.)          .014
FILE 7095,IN=PNC/LIBS        HAS BEEN OPENED FOR WRITING ON UNIT 18
CREATION TIME = 07/22/76    11:08:18
NTITL=    10
(TITL(I),I=1,NTITL)=    DECOMPOSITION MATRIX FOR SAMPLE CASE
UNIT 18 ENDFILED AND RELEASED
+:      USER COULD STOP HERE AND PERFORM
+:      REMAINING COMMANDS LATER, OR HE
+:      MAY CONTINUE.
+:      READ CONSTANT SUB-VECTOR FILES.
+:  BCRE
FILE 7068,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:08:54
UNIT  2 REWOUND AND RELEASED
FILE 7069,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:08:54
UNIT  2 REWOUND AND RELEASED
FILE 7070,BC=PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:08:54
UNIT  2 REWOUND AND RELEASED
```

FIGURE 8.3-2 Output for SOLN Code for Test Case No. 3 (Cont'd)

```
+:       READ DECOMPOSITION MATRIX FILE
+:       BACK IN,
+:  IVNR
FILE 7095,IN-PNC           HAS BEEN OPENED FOR READING ON UNIT   4
CREATION TIME = 07/22/76     11:08:18
MACH NO.=    .1234
UNIT   4 REWOUND AND RELEASED
+:       PERFORM FORWARD ELIMINATION AND
+:       BACKWARD SUBSTITUTION, AND DEFINE
+:       SOLUTION SUB-VECTOR OUTPUT FILES.
+:  MULT
(OUTA(I),OUTANT(I),I=1,NCMP)=
 7077     0
 7078     0
 7079     0


SOLUTION VECTOR NO.  1
----------------------

FILE 7077,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT   3
CREATION TIME = 07/22/76     11:08:23

UNIT NO,   3
COMPONENT NO.   1  SOLUTIONS:
       .1000000E+01   .2000000E+01   .3000000E+01   .4000000E+01   .5000000E+01
UNIT   3 ENDFILED AND RELEASED
FILE 7078,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT   3
CREATION TIME = 07/22/76     11:08:24

UNIT NO,   3
COMPONENT NO.   2  SOLUTIONS:
       .6000000E+01   .7000000E+01   .8000000E+01
UNIT   3 ENDFILED AND RELEASED
FILE 7079,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT   3
CREATION TIME = 07/22/76     11:08:24

UNIT NO,   3
COMPONENT NO,   3  SOLUTIONS:
       .9000001E+01   .1000000E+02
UNIT   3 ENDFILED AND RELEASED
```

FIGURE 8.3-2 Output for SCLN Code for Test Case No. 3 (Cont'd)

```
SOLUTION VECTOR NO. 2
***********************

FILE 7077,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO. 3
COMPONENT NO.   1  SOLUTIONS:
        .2000000E+01   .4000000E+01   .6000000E+01   .8000001E+01   .9999999E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7078,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO. 3
COMPONENT NO.   2  SOLUTIONS:
        .1200000E+02   .1400000E+02   .1600000E+02
UNIT  3 ENDFILED AND RELEASED
FILE 7079,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3

UNIT NO. 3
COMPONENT NO.   3  SOLUTIONS:
        .1800000E+02   .2000000E+02
UNIT  3 ENDFILED AND RELEASED
+I      ALL DONE.
+I   STOP
 STOP 777
```

FIGURE 8.3-2 Output for SCLN Code for Test Case No. 3 (Concluded)

```
TRVE
   SAMPLE CASE 4 FOR SOLN CODE.
   SOLUTION BY ITERATION PROCEDURE
   WITH LU DECOMPOSITION OF DIAGONAL
   BLOCKS, READING IN OF ONE DECOMP.
   OSED DIAGONAL BLOCK AND A PRE.
   DETERMINED GUESS AT ONE SOLUTION
   SUB.VECTOR, AND SLIGHT OVER.
   RELAXATION.
   BATCH MODE.
   THREE AIRCRAFT COMPONENTS.
CNTL
   1
   READ DIAGONAL INFLUENCE SUB.MATRIX
   ASSOCIATED WITH SECOND AIRCRAFT COMPONENT.
VNRF
 7063
   PERFORM DECOMPOSITION OF THIS SUB.MATRIX
   AND STORE.
INVF
  0 7096     1
  17
   DECOMPOSITION OF DIAGONAL SUB.MATRIX
   FOR SECOND AIRCRAFT COMPONENT.
   USER COULD STOP HERE AND PERFORM
   SUBSEQUENT COMMANDS LATER.
   NOW DETERMINE SOLUTION FOR ALL
   THREE AIRCRAFT COMPONENTS.
CNTL
   3   1
   READ INFLUENCE SUB.MATRIX FILES,
   AND THE PRE.COMPUTED AND PRE.
   STORED DECOMPOSITION OF ONE OF
   THE DIAGONAL SUB.MATRICES.
VNRF
 7059
 7060
 7061
 7062
 7096       .1
 7064
 7065
 7066
 7067
   THE CUMM COMMAND IS NOT USED WITH
```

FIGURE 8.4-1 Input for SCLN Code for Test Case No. 4

```
      THE ITERATION PROCEDURE.
      READ CONSTANT SUB-VECTOR FILES.
BCRE
 7068
 7069
 7070
      SOLVE DIAGONAL BLOCKS BY LU
      DECOMPOSITION, DEFINE THE
      SOLUTION SUB-VECTOR FILES, THE
      PRE-DETERMINED APPROXIMATE
      SOLUTION SUB-VECTOR FILES,
      AMOUNT OF OVER-RELAXATION, AND
      AMOUNT OF INTERMEDIATE PRINTOUT.
SOLV
      0    1
 7080
 7081
 7082
0
      3 7073
     10
1.05
      5
      ALL DONE.
STOP
```

FIGURE 8.4-1 Input for SCLN Code for Test Case No. 4 (Concluded)

```
POTFAN EQUATION SOLVING PROGRAM (SOLN). VERSION 1.1
*************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76    11:08:38

+:      SAMPLE CASE 4 FOR SOLN CODE.
+:      SOLUTION BY ITERATION PROCEDURE
+:      WITH LU DECOMPOSITION OF DIAGONAL
+:      BLOCKS, READING IN OF ONE DECOMP-
+:      OSED DIAGONAL BLOCK AND A PRE-
+:      DETERMINED GUESS AT ONE SOLUTION
+:      SUB-VECTOR, AND SLIGHT OVER-
+:      RELAXATION.
+:      BATCH MODE.
+:      THREE AIRCRAFT COMPONENTS.
+:  CNTL
NCMP=     1  ITERAT=     0
+:      READ DIAGONAL INFLUENCE SUB-MATRIX
+:      ASSOCIATED WITH SECOND AIRCRAFT COMPONENT.
+:  VNRE
FILE 7063,VN-PNC          HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
+:      PERFORM DECOMPOSITION OF THIS SUB-MATRIX
+:      AND STORE.
+:  INVE
METHOD=    4  MTINVW= 7096  IDC=     1
ENTER SUB. PART   #FRAGS=   2    INV= -2
MATRIX LOADED FROM UNIT  13  USING    2  READS AND    0   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

***  MATRIX INVERSE DEMANDED

ENTER SUB. TRIP.  N=    3
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4

```
EXIT TRIP
            INVERSION OR DECOMPOSITION TIME (SEC.)          .007
FILE 7096,IN-PNC/LIB$       HAS BEEN OPENED FOR WRITING ON UNIT 18
CREATION TIME = 07/22/76     11:08:41
NTITL=    17
(TITL(I),I=1,NTITL)=     DECOMPOSITION OF DIAGONAL SUB-MATRIX
UNIT 18 ENDFILED AND RELEASED
+:       FOR SECOND AIRCRAFT COMPONENT.
+:       USER COULD STOP HERE AND PERFORM
+:       SUBSEQUENT COMMANDS LATER.
+:       NOW DETERMINE SOLUTION FOR ALL
+:       THREE AIRCRAFT COMPONENTS.
+: CNTL
NCMP=     3  ITERAT=     1
+:       READ INFLUENCE SUB-MATRIX FILES,
+:       AND THE PRE-COMPUTED AND PRE-
+:       STORED DECOMPOSITION OF ONE OF
+:       THE DIAGONAL SUB-MATRICES.
+: VNRE
FILE 7059,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7060,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7061,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7062,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7096,IN-PNC             HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME = 07/22/76     11:08:41
UNIT  4 REWOUND AND RELEASED
FILE 7064,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7065,VN-PNC             HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Cont'd)

```
UNIT   1 REWOUND AND RELEASED
FILE 7066,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT   1 REWOUND AND RELEASED
FILE 7067,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT   1 REWOUND AND RELEASED
+:       THE COMB COMMAND IS NOT USED WITH
+:       THE ITERATION PROCEDURE.
+:       READ CONSTANT SUB-VECTOR FILES.
+:  RCPE
FILE 7068,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
FILE 7069,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
FILE 7070,RC-PNC              HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:08:54
UNIT   2 REWOUND AND RELEASED
+:       SOLVE DIAGONAL BLOCKS BY LU
+:       DECOMPOSITION, DEFINE THE
+:       SOLUTION SUB-VECTOR FILES, THE
+:       PRE-DETERMINED APPROXIMATE
+:       SOLUTION SUB-VECTOR FILES,
+:       AMOUNT OF OVER-RELAXATION, AND
+:       AMOUNT OF INTERMEDIATE PRINTOUT.
+:  SOLV
METHOD=
IPS=      1
IPRINT=
(OUTA,OUTANT,I=1,NCMP)=
 7080     0
 7081     0
 7082     0
(INPS,INPSNT,I=1,NCMP)=
    0     0
    0     0
 7073     0
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Cont'd)

```
MAXITI=   10
OMEGA=1.050
ITERP=    5

ENTER SUB. TRIP.  N=    5
EXIT TRIP
FILE 7096,IN-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76     11:08:41
UNIT  4 REWOUND AND RELEASED

ENTER SUB. TRIP.  N=    2
EXIT TRIP

SOLUTION VECTOR NO.  1
====================

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

INITIAL GUESS

FILE 7068,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7069,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7070,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT      1
  .31153E+08  .54017E+08  .78976E+08  .10474E+09  .13062E+09

COMPONENT NUMBER     1
NUMBER OF ELEMENTS=    5
SOLUTIONS
  .99872E+00  .19509E+01  .28292E+01  .35730E+01  .10571E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT    2
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Cont'd)

```
  .61007E.08  .26072E+08  .99609E+07

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     2
  .15588E.09  .17886F+09  .19559F+09

COMPONENT NUMBER     2
NUMBER OF ELEMENTS=     3
SOLUTIONS
  .31884E.01  .67366F+01  .14672F+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     3
  .32890F.07  .89082F+06

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     3
  .86929F.08  .37210F+08

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     3
  .19748F.07  .16773F+09
FILE 7073,SP-PNC          HAS BEEN OPENED FOR READING ON-UNIT  4
CREATION TIME=07/22/76    11:06:24
UNIT  4 REWOUND AND RELEASED

COMPONENT NUMBER     3
NUMBER OF ELEMENTS=     2
SOLUTIONS
  .90000F.01  .10000F+02

ITERATION     1  LARGEST RELATIVE ERROR=     .626337E+00

ITERATION     2  LARGEST RELATIVE ERROR=     .974581F-01

ITERATION     3  LARGEST RELATIVE ERROR=     .132704F-01

ITERATION     4  LARGEST RELATIVE ERROR=     .247379E-02

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT
```

FIGURE 8.4-2 Output for SCLN Code for Test Case No. 4 (Cont'd)

```
ITERATION    5  WITH   10  ALLOWABLE

 INFLUENCE OF AIRCRAFT COMPONENT      2  ON AIRCRAFT   COMPONENT    1
   .60085E+06  .23572E+07  .75248E+07  .20585E+08  .49960E+08

 INFLUENCE OF AIRCRAFT COMPONENT      3  ON AIRCRAFT   COMPONENT    1
   .11621E+04  .20964E+05  .16933E+06  .86700E+06  .33008E+07

 COMPONENT NUMBER     1
 NUMBER OF ELEMENTS=    5
 SOLUTIONS
   .10000E+01  .20000E+01  .30000E+01  .40000E+01  .50005E+01

 INFLUENCE OF AIRCRAFT COMPONENT      1  ON AIRCRAFT   COMPONENT    2
   .35414E+08  .14793E+08  .55072E+07

 INFLUENCE OF AIRCRAFT COMPONENT      3  ON AIRCRAFT   COMPONENT    2
   .10212E+08  .27111E+08  .64020E+08

 COMPONENT NUMBER     2
 NUMBER OF ELEMENTS=    3
 SOLUTIONS
   .59998E+01  .70000E+01  .79999E+01

 INFLUENCE OF AIRCRAFT COMPONENT      1  ON AIRCRAFT   COMPONENT    3
   .17660E+07  .46302E+06

 INFLUENCE OF AIRCRAFT COMPONENT      2  ON AIRCRAFT   COMPONENT    3
   .57884E+08  .24221E+08

 COMPONENT NUMBER     3
 NUMBER OF ELEMENTS=    2
 SOLUTIONS
   .90001E+01  .10000E+02

 ITERATION    5  LARGEST RELATIVE ERROR=    .470922E-03

 ITERATION    6  LARGEST RELATIVE ERROR=    .821186E-04
 FILE 7080,SO-PNC/LIB$     HAS BEEN OPENED FOR WRITING ON UNIT  3
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Cont'd)

```
CREATION TIME = 07/22/76    11:09:05
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   1  SOLUTIONS:
      .9999999E+00   .1999999F+01   .2999998E+01   .3999996F+01   .5000081E+01
FILE 7081,SC-PNC/LIBS      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:09:06
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   2  SOLUTIONS:
      .5999970F+01   .7000002F+01   .7999981E+01
FILE 7082,SO-PNC/LIBS      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:09:06
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   3  SOLUTIONS:
      .9000007F+01   .1000000F+02

NO. OF ITERATIONS TO CONVERGE=    6  LARGEST RELATIVE ERROR=    .821186E-04


SOLUTION VECTOR NO.  2
--------------------

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

INITIAL GUESS

FILE 7068,BC-PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7069,BC-PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7070,BC-PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
```

FIGURE 8.4-2 Output for SCLN Code for Test Case No. 4 (Cont'd)

```
INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     1
  .62307E+08  .10803E+09  .15795E+09  .20947E+09  .26123E+09

COMPONENT NUMBER     1
NUMBER OF ELEMENTS=     5
SOLUTIONS
  .19974E+01  .39019E+01  .56584E+01  .71460E+01  .21142E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     2
  .12201E+09  .52144E+08  .19922E+08

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     2
  .31176E+09  .35773E+09  .39118E+09

COMPONENT NUMBER     2
NUMBER OF ELEMENTS=     3
SOLUTIONS
  .63767E+01  .13473E+02  .29344E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     3
  .65781E+07  .17816E+07

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     3
  .17386E+09  .74420E+08

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     3
  .39496E+09  .33546E+09
FILE 7073,SO=PNC          HAS BEEN OPENED FOR READING ON UNIT   4
CREATION TIME=07/22/76    11:06:24
UNIT  4 REWOUND AND RELEASED

COMPONENT NUMBER     3
NUMBER OF ELEMENTS=     2
SOLUTIONS
  .18000E+02  .20000E+02
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Cont'd)

```
ITERATION    1  LARGEST RELATIVE ERROR=    .626337E+00

ITERATION    2  LARGEST RELATIVE ERROR=    .974581E-01

ITERATION    3  LARGEST RELATIVE ERROR=    .132704E-01

ITERATION    4  LARGEST RELATIVE ERROR=    .247379E-02

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

ITERATION    5  WITH   10  ALLOWABLE

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     1
  .12017E+07  .47145E+07  .15050E+08  .41169E+08  .99921E+08

INFLUENCE OF AIRCRAFT COMPONENT     3  ON AIRCRAFT   COMPONENT     1
  .23241E+04  .41928E+05  .33867E+06  .17340E+07  .66016E+07

COMPONENT NUMBER    1
NUMBER OF ELEMENTS=    5
SOLUTIONS
  .20000E+01  .40000E+01  .60000E+01  .80000E+01  .10001E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     2
  .70829E+08  .29587E+08  .11014E+08

INFLUENCE OF AIRCRAFT COMPONENT     3  ON AIRCRAFT   COMPONENT     2
  .20423E+08  .54221E+08  .12804E+09

COMPONENT NUMBER    2
NUMBER OF ELEMENTS=    3
SOLUTIONS
  .12000E+02  .14000E+02  .16000E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     3
  .35319E+07  .92603E+06

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     3
  .11577E+09  .48443E+08
```

FIGURE 8.4-2 Output for SCLN Code for Test Case No. 4 (Cont'd)

```
COMPONENT NUMBER     3
NUMBER OF ELEMENTS=     2
SOLUTIONS
   .18000F+02  .20000F+02

ITERATION     5  LARGEST RELATIVE ERROR=     .470922E-03

ITERATION     6  LARGEST RELATIVE ERROR=     .821186E-04
FILE 7080,SO-PNC/LIB$     HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   1  SOLUTIONS:
       .2000000E+01  .3999999F+01  .5999997E+01  .7999992F+01  .1000016F+02
FILE 7081,SO-PNC/LIB$     HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   2  SOLUTIONS:
       .1199994E+02  .1400000E+02  .1599996E+02
FILE 7082,SO-PNC/LIB$     HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   3  SOLUTIONS:
       .1800001E+02  .2000000F+02

NO. OF ITERATIONS TO CONVERGE=    6  LARGEST RELATIVE ERROR=    .821186E-04
+!      ALL DONE.
+!  STOP
 STOP 777
```

FIGURE 8.4-2 Output for SOLN Code for Test Case No. 4 (Concluded)

```
TRUE
     SAMPLE CASE 5 FOR SOLN CODE.
     SOLUTION BY A TWO STEP PROCEDURE.
     FIRST, TWO AIRCRAFT COMPONENTS ARE SOLVED
     BY DIRECT LU DECOMPOSITION. THEN, THE
     INFLUENCE OF A THIRD AIRCRAFT COMPONENT
     ON THE FIRST TWO IS COMPUTED BY THE
     ITERATION PROCEDURE. THE DIRECT SOLUTIONS
     FOR THE FIRST TWO AIRCRAFT COMPONENTS ARE USED
     AS INITIAL GUESSES TO THE THREE COMPONENT
     ITERATION SOLUTION.
CNTL
     2
     READ INFLUENCE SUB-MATRIX FILES
     FOR THE FIRST TWO AIRCRAFT
     COMPONENTS.
VNRE
 7059
 7060
 7062
 7063
     SINCE ENTIRE MATRIX FITS IN CORE,
     COMB COMMAND IS NOT NEEDED.
     READ CONSTANT SUB-VECTOR FILES.
BCRE
 7068
 7069
     SOLVE BY LU DECOMPOSITION,
     AND DEFINE THE SOLUTION
     SUB-VECTOR OUTPUT FILES.
SOLV
     0
 7083
 7084
0
     USER COULD STOP HERE AND PERFORM
     REMAINING COMMANDS LATER, OR HE
     MAY CONTINUE.
     DETERMINE INFLUENCE OF THIRD
     AIRCRAFT COMPONENT VIA ITERATION
     PROCEDURE.
CNTL
     3    1
     REREAD ALL INFLUENCE SUB-MATRIX
     FILES.
```

FIGURE 8.5-1 Input for SOLN Code for Test Case No. 5

```
VNRE
  7059
  7060
  7061
  7062
  7063
  7064
  7065
  7066
  7067
     THE COMB COMMAND MAY NOT BE USED
     WITH THE ITERATION PROCEDURE.
     REREAD ALL CONSTANT SUB-VECTOR FILES.
BCRE
  7068
  7069
  7070
     SOLVE DIAGONAL BLOCKS BY LU
     DECOMPOSITION, DEFINE THE
     SOLUTION SUB-VECTOR FILES, THE
     PRE-DETERMINED APPROXIMATE
     SOLUTION SUB-VECTOR FILES,
     MAXIMUM NUMBER OF ITERATIONS,
     AMOUNT OF OVER-RELAXATION,
     AND AMOUNT OF INTERMEDIATE
     PRINTOUT.
SOLV
     0     2
  7085
  7086
  7087
0
     1 7083
     2 7084
    10
1,10
0
     ALL DONE.
STOP
```

FIGURE 8.5-1 Input for SOLN Code for Test Case No. 5 (Concluded)

```
POTFAN EQUATION SOLVING PROGRAM (SOLN).  VERSION 1.1
****************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76    11:09:44

+:      SAMPLE CASE 5 FOR SOLN CODE.
+:      SOLUTION BY A TWO STEP PROCEDURE.
+:      FIRST, TWO AIRCRAFT COMPONENTS ARE SOLVED
+:      BY DIRECT LU DECOMPOSITION.  THEN, THE
+:      INFLUENCE OF A THIRD AIRCRAFT COMPONENT
+:      ON THE FIRST TWO IS COMPUTED BY THE
+:      ITERATION PROCEDURE.  THE DIRECT SOLUTIONS
+:      FOR THE FIRST TWO AIRCRAFT COMPONENTS ARE USED
+:      AS INITIAL GUESSES TO THE THREE COMPONENT
+:      ITERATION SOLUTION.
+:  CNTL
NCMP=     2  ITERAT=     0
+:      READ INFLUENCE SUB-MATRIX FILES
+:      FOR THE FIRST TWO AIRCRAFT
+:      COMPONENTS.
+:  VNRE
FILE 7059,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7060,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7062,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7063,VN-PNC           HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:08:54
UNIT  1 REWOUND AND RELEASED
+:      SINCE ENTIRE MATRIX FITS IN CORE,
+:      COMP COMMAND IS NOT NEEDED.
+:      READ CONSTANT SUB-VECTOR FILES.
+:  BCRE
FILE 7068,BC-PNC           HAS BEEN OPENED FOR READING ON UNIT  2
```

FIGURE 8.5-2 Output for SCLN Code for Test Case No. 5

```
CREATION TIME=07/07/76    04:08:54
UNIT  2 REWOUND AND RELEASED
FILE 7069,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:08:54
UNIT  2 REWOUND AND RELEASED
+:       SOLVE BY LU DECOMPOSITION,
+:       AND DEFINE THE SOLUTION
+:       SUB-VECTOR OUTPUT FILES.
+:  SOLV
METHOD=     0
IPS=     0
IPRINT=     0
(OUTA,OUTANT,I=1,NCMP)=
 7083    0
 7084    0
ENTER SUB. PART   #FRAGS=  10     INV= -2
MATRIX LOADED FROM UNIT  13  USING   10  READS AND   30   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

ENTER SUB. TRIP.  N=    8
EXIT TRIP

OUTPUT
SOLUTION TIME (SEC.)      .011

SOLUTION VECTOR NO.  1
----------------------

FILE 7083,SO-PNC/LIBS     HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:09:55

UNIT NO.  3
COMPONENT NO.   1  SOLUTIONS:
      .1000553E+01   .2003342E+01   .3003688E+01   .3990381E+01   .4943312E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7084,SO-PNC/LIBS     HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:09:56

UNIT NO.  3
COMPONENT NO.   2  SOLUTIONS:
```

FIGURE 8.5-2 Output for SOLN Code for Test Case No. 5 (Cont'd)

```
       .5635605E+01    .6638698E+01    .1461401E+02
UNIT   3 ENDFILED AND RELEASED

OUTPUT
SOLUTION TIME (SEC.)          .003


SOLUTION VECTOR NO.  2
----------------------

FILE 7083,SO=PNC/LIB*        HAS BEEN OPENED FOR WRITING ON UNIT   3

UNIT NO.   3
COMPONENT NO.    1  SOLUTIONS:
       .2001107E+01    .4006683E+01    .6007377E+01    .7980762E+01    .9886623E+01
UNIT   3 ENDFILED AND RELEASED
FILE 7084,SO=PNC/LIB*        HAS BEEN OPENED FOR WRITING ON UNIT   3

UNIT NO.   3
COMPONENT NO.    2  SOLUTIONS:
       .1167121E+02    .1327740E+02    .2922803E+02
UNIT   3 ENDFILED AND RELEASED
+:       USER COULD STOP HERE AND PERFORM
+:       REMAINING COMMANDS LATER, OR HE
+:       MAY CONTINUE.
+:       DETERMINE INFLUENCE OF THIRD
+:       AIRCRAFT COMPONENT VIA ITERATION
+:       PROCEDURE.
+:  CNTL
NCMP=      3   ITERAT=      1
+:       REREAD ALL INFLUENCE SUB-MATRIX
+:       FILES.
+:  VNRE
FILE 7059,VN=PNC             HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT   1 REWOUND AND RELEASED
FILE 7060,VN=PNC             HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT   1 REWOUND AND RELEASED
FILE 7061,VN=PNC             HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:08:54
UNIT   1 REWOUND AND RELEASED
```

FIGURE 8.5-2 Output for SCLN Code for Test Case No. 5 (Cont'd)

```
FILE 7062,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7063,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7064,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7065,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7066,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
FILE 7067,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:08:54
UNIT  1 REWOUND AND RELEASED
+:      THE COMB COMMAND MAY NOT BE USED
+:      WITH THE ITERATION PROCEDURE.
+:      REREAD ALL CONSTANT SUB-VECTOR FILES.
+:  RCRE
FILE 7068,RC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:08:54
UNIT  2 REWOUND AND RELEASED
FILE 7069,RC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:08:54
UNIT  2 REWOUND AND RELEASED
FILE 7070,RC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:08:54
UNIT  2 REWOUND AND RELEASED
+:      SOLVE DIAGONAL BLOCKS BY LU
+:      DECOMPOSITION, DEFINE THE
+:      SOLUTION SUB-VECTOR FILES, THE
+:      PRE-DETERMINED APPROXIMATE
+:      SOLUTION SUB-VECTOR FILES,
+:      MAXIMUM NUMBER OF ITERATIONS,
+:      AMOUNT OF OVER-RELAXATION,
+:      AND AMOUNT OF INTERMEDIATE
+:      PRINTOUT.
+:  SOLV
METHOD=    0
```

FIGURE 8.5-2 Output for SOLN Code for Test Case No. 5 (Cont'd)

```
IPS=      2
IPRINT=      3
(OUTA,OUTANT,I=1,NCMP)=
 7085     0
 7086     0
 7087     0
(INPS,INPSNT,I=1,NCMP)=
 7083     0
 7084     0
    0     0
MAXIT1=    10
OMEGA=1.100
ITERP=     0.

ENTER SUB. TRIP.  N=     5
EXIT TRIP

ENTER SUB. TRIP.  N=     3
EXIT TRIP

ENTER SUB. TRIP.  N=     2
EXIT TRIP


SOLUTION VECTOR NO.  1
********************
FILE 7068,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT   4
UNIT  4 REWOUND AND RELEASED
FILE 7069,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT   4
UNIT  4 REWOUND AND RELEASED
FILE 7070,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT   4
UNIT  4 REWOUND AND RELEASED
FILE 7083,SO=PNC            HAS BEEN OPENED FOR READING ON UNIT   4
CREATION TIME=07/22/76     11:09:55
UNIT  4 REWOUND AND RELEASED
FILE 7084,SO=PNC            HAS BEEN OPENED FOR READING ON UNIT   4
CREATION TIME=07/22/76     11:09:56
UNIT  4 REWOUND AND RELEASED

ITERATION     1  LARGEST RELATIVE ERROR=     .476246E+00
```

FIGURE 8.5-2 Output for SOLN Code for Test Case No. 5 (Cont'd)

```
ITERATION     2  LARGEST RELATIVE ERROR=     .115208E+00

ITERATION     3  LARGEST RELATIVE ERROR=     .235574E-01

ITERATION     4  LARGEST RELATIVE ERROR=     .303637E-02

ITERATION     5  LARGEST RELATIVE ERROR=     .122435E-03

ITERATION     6  LARGEST RELATIVE ERROR=     .573020E-04
FILE 7085,SO-PNC/LIB$       HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:10:46
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   1  SOLUTIONS:
       .1000000E+01   .2000000E+01   .3000000E+01   .3999999E+01   .5000026E+01
FILE 7086,SO-PNC/LIB$       HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:10:47
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   2  SOLUTIONS:
       .5999997E+01   .7000000E+01   .8000011E+01
FILE 7087,SO-PNC/LIB$       HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:10:48
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   3  SOLUTIONS:
       .8999996E+01   .1000000E+02

NO. OF ITERATIONS TO CONVERGE=    6  LARGEST RELATIVE ERROR=     .573020E-04


SOLUTION VECTOR NO.  2
----------------------
FILE 7068,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7069,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7070,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
```

FIGURE 8.5-2 Output for SOLN Code for Test Case No. 5 (Cont'd)

```
UNIT  4 REWOUND AND RELEASED
FILE 7083,SO=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:09:55
UNIT  4 REWOUND AND RELEASED
FILE 7084,SO=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:09:56
UNIT  4 REWOUND AND RELEASED

ITERATION    1  LARGEST RELATIVE ERROR=    .476246E+00

ITERATION    2  LARGEST RELATIVE ERROR=    .115208E+00

ITERATION    3  LARGEST RELATIVE ERROR=    .235574E-01

ITERATION    4  LARGEST RELATIVE ERROR=    .303637E-02

ITERATION    5  LARGEST RELATIVE ERROR=    .122435E-03

ITERATION    6  LARGEST RELATIVE ERROR=    .573020E-04
FILE 7085,SO=PNC/LIB*      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   1  SOLUTIONS:
      .2000000E+01   .4000000E+01   .5999999E+01   .7999999E+01   .1000005E+02
FILE 7086,SO=PNC/LIB*      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   2  SOLUTIONS:
      .1199999E+02   .1400000E+02   .1600002E+02
FILE 7087,SO=PNC/LIB*      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   3  SOLUTIONS:
      .1799999E+02   .2000000E+02

NO. OF ITERATIONS TO CONVERGE=    6  LARGEST RELATIVE ERROR=    .573020E-04
+:     ALL DONE.
+:  STOP
 STOP 777
```

FIGURE 8.5-2 Output for SCLN Code for Test Case No. 5 (Concluded)

```
TRUE
    SAMPLE CASE 6 FOR SOLN CODE.
    SIMILAR TO SAMPLE CASE 4, EXCEPT
    THAT A RECTANGULAR MATRIX (17X10)
    IS SOLVED INSTEAD OF A SQUARE
    MATRIX. SOLUTION BY ITERATION
    PROCEDURE WITH INVERSE OF DIAGONAL
    BLOCKS, READING IN OF ONE INVERSE OF A
    DIAGONAL BLOCK AND A PRE-DETERMINED
    GUESS AT ONE SOLUTION SUB-VECTOR, AND
    SLIGHT OVER-RELAXATION.
    BATCH MODE.
    THREE AIRCRAFT COMPONENTS.
CNTL
    1
    READ DIAGONAL INFLUENCE SUB-MATRIX
    ASSOCIATED WITH SECOND AIRCRAFT COMPONENT.
VNRE
 7081
    PERFORM INVERSE OF THIS SUB-MATRIX
    AND STORE.
INVF
    1 7097
   17
    INVERSE OF DIAGONAL SUB-MATRIX
    FOR SECOND AIRCRAFT COMPONENT.
    USER COULD STOP HERE AND PERFORM
    SUBSEQUENT COMMANDS LATER.
    NOW DETERMINE SOLUTION FOR
    ALL THREE AIRCRAFT COMPONENTS.
CNTL
    3
    READ INFLUENCE SUB-MATRIX FILES
    AND THE PREVIOUSLY COMPUTED AND
    STORED INVERSE OF ONE OF THE DIAGONAL
    SUB-MATRICES.
VNRE
 7077
 7078
 7079
 7080
 7097          -1
 7082
 7083
 7084
```

FIGURE 8.6-1 Input for SOLN Code for Test Case No. 6

```
7085
    THE COMB COMMAND IS NOT USED WITH
    THE ITERATION PROCEDURE.
    READ CONSTANT SUB-VECTOR FILES.
BCRE
 7086
 7087
 7088
    COMPUTE INVERSE OF DIAGONAL BLOCKS BY
    HOUSEHOLDER PROCEDURE, DEFINE THE
    SOLUTION SUB-VECTOR FILES, THE
    PRE-DETERMINED APPROXIMATE SOLUTION
    SUB-VECTOR FILES, AMOUNT OF
    OVER-RELAXATION, AND AMOUNT OF
    INTERMEDIATE PRINTOUT.
SOLV
    1     1
 7088
 7089
 7090
 0
    3 7073
   10
1.05
    5
    ALL DONE.
STOP
```

FIGURE 8.6-1 Input for SCLN Code for Test Case No. 6 (Concluded)

```
POTFAN EQUATION SOLVING PROGRAM (SOLN).  VERSION 1.1
****************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76    11:12:49

+:       SAMPLE CASE 6 FOR SOLN CODE.
+:       SIMILAR TO SAMPLE CASE 4, EXCEPT
+:       THAT A RECTANGULAR MATRIX (17X10)
+:       IS SOLVED INSTEAD OF A SQUARE
+:       MATRIX.  SOLUTION BY ITERATION
+:       PROCEDURE WITH INVERSE OF DIAGONAL
+:       BLOCKS, READING IN OF ONE INVERSE OF A
+:       DIAGONAL BLOCK AND A PRE-DETERMINED
+:       GUESS AT ONE SOLUTION SUB-VECTOR, AND
+:       SLIGHT OVER-RELAXATION.
+:       BATCH MODE.
+:       THREE AIRCRAFT COMPONENTS.
+:  CNTL
NCMP=    1   ITERAT=    0
+:       READ DIAGONAL INFLUENCE SUB-MATRIX
+:       ASSOCIATED WITH SECOND AIRCRAFT COMPONENT.
+:  VNRE
FILE 7081'VN-PNC          HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:09:01
UNIT   1 REWOUND AND RELEASED
+:       PERFORM INVERSE OF THIS SUB-MATRIX
+:       AND STORE.
+:  INVE
METHOD=    1   NTINVW= 7097   IDC=     0
ENTER SUB. PART    #FRAGS=    2      INV= 01
MATRIX LOADED FROM UNIT   13  USING     2  READS AND     0    SKIPS
NO PARTITIONING REQUIRED
EXIT PART

***  MATRIX INVERSE DEMANDED

ENTER SUB. HREC.       6 ROWS       3 COLUMNS.    MATRIX ON UNIT 11
EXIT HREC.  INVERSE ON UNIT 11
```

FIGURE 8.6-2 Output for SOLN Code for Test Case No. 6

```
            INVERSION OR DECOMPOSITION TIME ,(SEC.)          .145
FILE 7097,IN-PNC/LIB8      HAS BEEN OPENED FOR WRITING ON UNIT 18
CREATION TIME = 07/22/76     11:13:12
NTITL =    17
(TITL(I),I=1,NTITL)=     INVERSE OF DIAGONAL SUB-MATRIX
UNIT 18 ENDFILED AND RELEASED
+:        FOR SECOND AIRCRAFT COMPONENT.
+:        USER COULD STOP HERE AND PERFORM
+:        SUBSEQUENT COMMANDS LATER.
+:        NOW DETERMINE SOLUTION FOR
+:        ALL THREE AIRCRAFT COMPONENTS.
+:   CNTL
NCMP=     3  ITERAT=     1
+:        READ INFLUENCE SUB-MATRIX FILES
+:        AND THE PREVIOUSLY COMPUTED AND
+:        STORED INVERSE OF ONE OF THE DIAGONAL
+:        SUB-MATRICES.
+:   VNRE
FILE 7077,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7078,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7079,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7080,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7097,IN-PNC              HAS BEEN OPENED FOR READING ON UNIT   4
CREATION TIME = 07/22/76     11:13:12
UNIT  4 REWOUND AND RELEASED
FILE 7082,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7083,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7084,VN-PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
```

FIGURE 8.6-2 Output for SCLN Code for Test Case No. 6 (Cont'd)

```
FILE 7085,VN=PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76      04:09:01
UNIT  1 REWOUND AND RELEASED
+:      THE COMB COMMAND IS NOT USED WITH
+:      THE ITERATION PROCEDURE.
+:      READ CONSTANT SUB-VECTOR FILES.
+:  BCRE
FILE 7086,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7087,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7088,BC=PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76      04:09:01
UNIT  2 REWOUND AND RELEASED
+:      COMPUTE INVERSE OF DIAGONAL BLOCKS BY
+:      HOUSEHOLDER PROCEDURE, DEFINE THE
+:      SOLUTION SUB-VECTOR FILES, THE
+:      PRE-DETERMINED APPROXIMATE SOLUTION
+:      SUB-VECTOR FILES, AMOUNT OF
+:      OVER-RELAXATION, AND AMOUNT OF
+:      INTERMEDIATE PRINTOUT.
+:  SOLV
METHOD=      1
IPS=     1
IPRINT=      0
(OUTA,OUTANT,I=1,NCMP)=
 7088     0
 7089     0
 7090     0
(INPS,INPSNT,I=1,NCMP)=
    0     0
    0     0
 7073     0
MAXIT1=    10
OMEGA=1,000
ITERP=     5

***  MATRIX INVERSE DEMANDED

ENTER SUB. HREC.      7 ROWS      5 COLUMNS.   MATRIX ON UNIT 12
```

FIGURE 8.6-2 Output for SOLN Code for Test Case No. 6 (Cont'd)

```
EXIT HREC. INVERSE ON UNIT 12

SUB. ATR TRANSPOSED A    7-BY-    5  MATRIX
FILE 7097,IN-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:13:12
UNIT  4 REWOUND AND RELEASED

SUB. ATR TRANSPOSED A    6-BY-    3  MATRIX

***  MATRIX INVERSE DEMANDED

ENTER SUB. HREC.     4 ROWS     2 COLUMNS.   MATRIX ON UNIT 11
EXIT HREC. INVERSE ON UNIT 12

SUB. ATR TRANSPOSED A    4-BY-    2  MATRIX


SOLUTION VECTOR NO. 1
---------------------

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

INITIAL GUESS

FILE 7086,BF-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7087,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7088,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     1
  .16837E+10  .23546E+10  .29693E+10  .35944E+10  .42939E+10  .45640E+10  .51133

COMPONENT NUMBER    1
NUMBER OF ELEMENTS=    5
SOLUTIONS
  .10189E+01  .18522E+01  .32019E+01  .24751E+01  .11417E+02

INFLUENCE OF AIRCRAFT COMPONENT    1  ON AIRCRAFT  COMPONENT   2
```

FIGURE 8.6-2 Output for SCLN Code for Test Case No. 6 (Cont'd)

```
 .31995E+10  .19929E+10  .12011E+10  .69707E+09  .38721E+09  .20430E+09

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     2
 .54056E+10  .56800E+10  .60713E+10  .61247E+10  .64752E+10  .63992E+10

COMPONENT NUMBER     2
NUMBER OF ELEMENTS=      3
SOLUTIONS
 .25062E+01  .58477E+01  .14991E+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT     3
 .10139E+09  .46712E+08  .19627E+08  .73323E+07

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     3
 .35849E+10  .22140E+10  .13208E+10  .75711E+09

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT     3
 .64604E+10  .63447E+10  .57012E+10  .55520E+10
FILE 7073,80-PNC        HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11106124
UNIT  4 REWOUND AND RELEASED

COMPONENT NUMBER     3
NUMBER OF ELEMENTS=      2
SOLUTIONS
 .90000E+01  .10000E+02

ITERATION     1  LARGEST RELATIVE ERROR=    .678650E+00

ITERATION     2  LARGEST RELATIVE ERROR=    .195333E+00

ITERATION     3  LARGEST RELATIVE ERROR=    .620144E-01

ITERATION     4  LARGEST RELATIVE ERROR=    .184026E-01

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

ITERATION     5  WITH   10  ALLOWABLE

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT     1
```

FIGURE 8.6-2 Output for SOLN Code for Test Case No. 6 (Cont'd)

```
 .54266E+08  .11454E+09  .22641E+09  .42344E+09  .75521E+09  .12927E+10  .21347

INFLUENCE OF AIRCRAFT COMPONENT    3  ON AIRCRAFT   COMPONENT    1
 .52388E+05  .31323E+06  .13171E+07  .43967E+07  .12442E+08  .31062E+08  .70246

COMPONENT NUMBER    1
NUMBER OF ELEMENTS=    5
SOLUTIONS
 .10000E+01  .19998E+01  .30004E+01  .39979E+01  .50104E+01

INFLUENCE OF AIRCRAFT COMPONENT    1  ON AIRCRAFT   COMPONENT    2
 .18394E+10  .11290E+10  .66956E+09  .38180E+09  .20804E+09  .10750E+09

INFLUENCE OF AIRCRAFT COMPONENT    3  ON AIRCRAFT   COMPONENT    2
 .14663E+09  .28644E+09  .52925E+09  .93262E+09  .15777E+10  .25758E+10

COMPONENT NUMBER    2
NUMBER OF ELEMENTS=    3
SOLUTIONS
 .59942E+01  .70017E+01  .79989E+01

INFLUENCE OF AIRCRAFT COMPONENT    1  ON AIRCRAFT   COMPONENT    3
 .52150E+08  .23448E+08  .95997E+07  .34910E+07

INFLUENCE OF AIRCRAFT COMPONENT    2  ON AIRCRAFT   COMPONENT    3
 .23319E+10  .14182E+10  .83225E+09  .46879E+09

COMPONENT NUMBER    3
NUMBER OF ELEMENTS=    2
SOLUTIONS
 .90004E+01  .99999E+01

ITERATION    5  LARGEST RELATIVE ERROR=    .529720E-02

ITERATION    6  LARGEST RELATIVE ERROR=    .149742E-02

ITERATION    7  LARGEST RELATIVE ERROR=    .419913E-03

ITERATION    8  LARGEST RELATIVE ERROR=    .117206E-03

ITERATION    9  LARGEST RELATIVE ERROR=    .326147E-04
FILE 7088.SO-PNC/LIB*     HAS BEEN OPENED FOR WRITING ON UNIT  3
```

FIGURE 8.6-2 Output for SCLN Code for Test Case No. 6 (Cont'd)

```
CREATION TIME = 07/22/76    11:14:45
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   1  SOLUTIONS:
        .1000000E+01   .1999999E+01   .3000002E+01   .3999987E+01   .5000062E+01
FILE 7089,SO=PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:14:52
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   2  SOLUTIONS:
        .5999966E+01   .7000009E+01   .7999996E+01
FILE 7090,SO=PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:14:53
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   3  SOLUTIONS:
        .9000001E+01   .9999999E+01

NO. OF ITERATIONS TO CONVERGE=    9  LARGEST RELATIVE ERROR=    .326147E-04


SOLUTION VECTOR NO.  2
====================

ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

INITIAL GUESS

FILE 7086,BC=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7087,BC=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7088,BC=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT   1
 .33673E+10  .47091E+10  .59387E+10  .71888E+10  .85877E+10  .91280E+10  .10227E+1
```

FIGURE 8.6-2 Output for SOLN Code for Test Case No. 6 (Cont'd)

```
COMPONENT NUMBER     1
NUMBER OF ELEMENTS=    5
SOLUTIONS
  .20379E+01  .37044E+01  .64038F+01  .49502E+01  .22833F+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT    2
  .63991F+10  .39858E+10  .24023E+10  .13941F+10  .77441F+09  .40860F+09

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT    2
  .10811F+11  .11360E+11  .12143F+11  .12249E+11  .12950E+11  .12798F+11

COMPONENT NUMBER     2
NUMBER OF ELEMENTS=    3
SOLUTIONS
  .50123E+01  .11695F+02  .29982F+02

INFLUENCE OF AIRCRAFT COMPONENT     1  ON AIRCRAFT   COMPONENT    3
  .20277E+09  .93423E+08  .39253E+08  .14665F+08

INFLUENCE OF AIRCRAFT COMPONENT     2  ON AIRCRAFT   COMPONENT    3
  .71697E+10  .44279F+10  .26415F+10  .15142F+10

INFLUENCE OF UNIFORM AND NON-UNIFORM FREESTREAM
AND ROTATION RATE VECTOR ON AIRCRAFT COMPONENT    3
  .12921E+11  .12689F+11  .11402F+11  .11104E+11
FILE 7073,SO-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:06:24
UNIT  4 REWOUND AND RELEASED

COMPONENT NUMBER     3
NUMBER OF ELEMENTS=    2
SOLUTIONS
  .18000F+02  .20000E+02

ITERATION     1  LARGEST RELATIVE ERROR=    .678650E+00

ITERATION     2  LARGEST RELATIVE ERROR=    .195333E+00

ITERATION     3  LARGEST RELATIVE ERROR=    .620144E-01

ITERATION     4  LARGEST RELATIVE ERROR=    .184026E-01
```

FIGURE 8.6-2 Output for SCLN Code for Test Case No. 6 (Cont'd)

```
ITERATION SOLUTION PROCEDURE INTERMEDIATE PRINTOUT

ITERATION     5  WITH   10  ALLOWABLE

INFLUENCE OF AIRCRAFT COMPONENT      2  ON AIRCRAFT   COMPONENT     1
  .10853E+09  .22908E+09  .45283E+09  .84688E+09  .15104E+10  .25854E+10  .42695E+10

INFLUENCE OF AIRCRAFT COMPONENT      3  ON AIRCRAFT   COMPONENT     1
  .10478E+06  .62647E+06  .26341E+07  .87933E+07  .24884E+08  .62124E+08  .14049E+09

COMPONENT NUMBER      1
NUMBER OF ELEMENTS=     5
SOLUTIONS
  .20000E+01  .39996E+01  .60008E+01  .79958E+01  .10021E+02

INFLUENCE OF AIRCRAFT COMPONENT      1  ON AIRCRAFT   COMPONENT     2
  .36788E+10  .22579E+10  .13391E+10  .76360E+09  .41609E+09  .21499E+09

INFLUENCE OF AIRCRAFT COMPONENT      3  ON AIRCRAFT   COMPONENT     2
  .29325E+09  .57288E+09  .10585E+10  .18692E+10  .31554E+10  .51517E+10

COMPONENT NUMBER      2
NUMBER OF ELEMENTS=     3
SOLUTIONS
  .11988E+02  .14003E+02  .15998E+02

INFLUENCE OF AIRCRAFT COMPONENT      1  ON AIRCRAFT   COMPONENT     3
  .10430E+09  .46895E+08  .19199E+08  .69820E+07

INFLUENCE OF AIRCRAFT COMPONENT      2  ON AIRCRAFT   COMPONENT     3
  .46639E+10  .28364E+10  .16645E+10  .93757E+09

COMPONENT NUMBER      3
NUMBER OF ELEMENTS=     2
SOLUTIONS
  .18001E+02  .20000E+02

ITERATION     5  LARGEST RELATIVE ERROR=     .529720E-02

ITERATION     6  LARGEST RELATIVE ERROR=     .149742E-02

ITERATION     7  LARGEST RELATIVE ERROR=     .419913E-03
```

FIGURE 8.6-2 Output for SOLN Code for Test Case No. 6 (Cont'd)

```
ITERATION    8  LARGEST RELATIVE ERROR=      .117206E-03

ITERATION    9  LARGEST RELATIVE ERROR=      .326147E-04
FILE 7088,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   1  SOLUTIONS:
      .2000000E+01   .3999999E+01   .6000004E+01   .7999975E+01   .1000013E+02
FILE 7089,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   2  SOLUTIONS:
      .1199993E+02   .1400002E+02   .1599999E+02
FILE 7090,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   3  SOLUTIONS:
      .1800000E+02   .2000000E+02

NO, OF ITERATIONS TO CONVERGE=    9  LARGEST RELATIVE ERROR=     .326147E-04
+:     ALL DONE.
+:  STOP
 STOP 777
```

FIGURE 8.6-2 Output for SCLN Code for Test Case No. 6 (Concluded)

```
TRUE
    SAMPLE CASE 7 FOR SOLN CODE.
    SIMILAR TO SAMPLE CASE 5 EXCEPT
    THAT A RECTANGULAR MATRIX (17X10) IS
    SOLVED INSTEAD OF A SQUARE MATRIX (10X10).
    SOLUTION BY A TWO STEP PROCEDURE.
    FIRST, TWO AIRCRAFT COMPONENTS ARE SOLVED
    BY DIRECT HOUSEHOLDER PROCEDURE.  THEN, THE
    INFLUENCE OF A THIRD AIRCRAFT COMPONENT
    ON THE FIRST TWO IS COMPUTED BY THE
    ITERATION PROCEDURE.  THE DIRECT SOLUTIONS
    FOR THE FIRST TWO AIRCRAFT COMPONENTS ARE USED
    AS INITIAL GUESSES TO THE THREE COMPONENT
    ITERATION SOLUTION.
CNTL
    2
    READ INFLUENCE SUB-MATRIX FILES
    FOR THE FIRST TWO AIRCRAFT
    COMPONENTS.
VNRE
 7077
 7078
 7080
 7081
    SINCE ENTIRE MATRIX FITS IN CORE,
    COMB COMMAND IS NOT NEEDED.
    READ CONSTANT SUB-VECTOR FILES.
BCRE
 7086
 7087
    SOLVE BY HOUSEHOLDER PROCEDURE,
    AND DEFINE THE SOLUTION
    SUB-VECTOR OUTPUT FILES.
SOLV
    1     0
 7091
 7092
0
    USER COULD STOP HERE AND PERFORM
    REMAINING COMMANDS LATER, OR HE
    MAY CONTINUE.
    DETERMINE INFLUENCE OF THIRD
    AIRCRAFT COMPONENT VIA ITERATION
    PROCEDURE.
CNTL
```

FIGURE 8.7-1 Input for SOLN Code for Test Case No. 7

```
      3     1
      REREAD ALL INFLUENCE SUB-MATRIX
      FILES.
 VNRE
  7077
  7078
  7079
  7080
  7081
  7082
  7083
  7084
  7085
      THE COMP COMMAND MAY NOT BE USED
      WITH THE ITERATION PROCEDURE.
      REREAD ALL CONSTANT SUB-VECTOR FILES.
 BCRE
  7086
  7087
  7088
      SOLVE DIAGONAL BLOCKS BY
      HOUSEHOLDER PROCEDURE, DEFINE THE
      SOLUTION SUB-VECTOR FILES, THE
      PRE-DETERMINED APPROXIMATE
      SOLUTION SUB-VECTOR FILES,
      MAXIMUM NUMBER OF ITERATIONS,
      AMOUNT OF OVER-RELAXATION,
      AND AMOUNT OF INTERMEDIATE
      PRINTOUT.
 SOLV
      1     2
  7093
  7094
  7095
 0
      1 7091
      2 7092
     10
 1.10
 0
      ALL DONE.
 STOP
```

FIGURE 8.7-1 Input for SCLN Code for Test Case No. 7 (Concluded)

```
POTFAN EQUATION SOLVING PROGRAM (SOLN). VERSION 1.1
****************************************************

DYNAMIC MEMORY = 10000

TIME =   07/22/76     11:16:17

+:       SAMPLE CASE 7 FOR SOLN CODE.
+:       SIMILAR TO SAMPLE CASE 5 EXCEPT
+:       THAT A RECTANGULAR MATRIX (17X10) IS
+:       SOLVED INSTEAD OF A SQUARE MATRIX (10X10).
+:       SOLUTION BY A TWO STEP PROCEDURE.
+:       FIRST, TWO AIRCRAFT COMPONENTS ARE SOLVED
+:       BY DIRECT HOUSEHOLDER PROCEDURE. THEN, THE
+:       INFLUENCE OF A THIRD AIRCRAFT COMPONENT
+:       ON THE FIRST TWO IS COMPUTED BY THE
+:       ITERATION PROCEDURE. THE DIRECT SOLUTIONS
+:       FOR THE FIRST TWO AIRCRAFT COMPONENTS ARE USED
+:       AS INITIAL GUESSES TO THE THREE COMPONENT
+:       ITERATION SOLUTION.
+:  CNTL
NCMP=     2  ITERAT=     0
+:       READ INFLUENCE SUB-MATRIX FILES
+:       FOR THE FIRST TWO AIRCRAFT
+:       COMPONENTS.
+:  .VNRE
FILE 7077,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:09:01
UNIT   1 REWOUND AND RELEASED
FILE 7078,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:09:01
UNIT   1 REWOUND AND RELEASED
FILE 7080,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:09:01
UNIT   1 REWOUND AND RELEASED
FILE 7081,VN=PNC              HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76    04:09:01
UNIT   1 REWOUND AND RELEASED
+:       SINCE ENTIRE MATRIX FITS IN CORE,
+:       COMB COMMAND IS NOT NEEDED.
```

FIGURE 8.7-2 Output for SOLN Code for Test Case No. 7

```
+1        READ CONSTANT SUB-VECTOR FILES.
+1  BCRE
FILE 7086,BC-PNC             HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7087,BC-PNC             HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76     04:09:01
UNIT  2 REWOUND AND RELEASED
+1        SOLVE BY HOUSEHOLDER PROCEDURE,
+1        AND DEFINE THE SOLUTION
+1        SUB-VECTOR OUTPUT FILES.
+1  SOLV
METHOD=     1
IPS=      0
IPRINT=     0
(OUTA,OUTANT,I=1,NCMP)=
 7091      0
 7092      0
ENTER SUB. PART   #FRAGS=  10      INV= -1
MATRIX LOADED FROM UNIT  13  USING   10  READS AND   30   SKIPS
NO PARTITIONING REQUIRED
EXIT PART

ENTER SUB. HOUSE.     13 ROWS      8 COLUMNS   2 SOLUTIONS
EXIT HOUSE

OUTPUT
SOLUTION TIME (SEC.)          .025

***  SOLUTION TIME IS THAT FOR ALL SOLUTIONS

SOLUTION VECTOR NO.  1
----------------------

FILE 7091,SO-PNC/LIB$      HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76     11:16:43

UNIT NO.  3
COMPONENT NO.   1  SOLUTIONS:
      .9983662E+00   .2007925E+01   .2984845E+01   .4071823E+01   .4751636E+01
UNIT  3 ENDFILED AND RELEASED
```

FIGURE 8.7-2 Output for SOLN Code for Test Case No. 7 (Cont'd)

```
FILE 7092,SO-PNC/LIB$          HAS BEEN OPENED FOR WRITING ON UNIT   3
CREATION TIME = 07/22/76      11:16:44

UNIT NO.  3
COMPONENT NO.    2  SOLUTIONS:
        .6593610E+01   .4802652E+01   .1518873E+02
UNIT  3 ENDFILED AND RELEASED

SOLUTION VECTOR NO.  2
**********************

FILE 7091,SO-PNC/LIB$          HAS BEEN OPENED FOR WRITING ON UNIT   3

UNIT NO.  3
COMPONENT NO.    1  SOLUTIONS:
        .1996732E+01   .4015849E+01   .5969690E+01   .8143645E+01   .9503272E+01
UNIT  3 ENDFILED AND RELEASED
FILE 7092,SO-PNC/LIB*          HAS BEEN OPENED FOR WRITING ON UNIT   3

UNIT NO.  3
COMPONENT NO.    2  SOLUTIONS:
        .1318722E+02   .9605304E+01   .3037746E+02
UNIT  3 ENDFILED AND RELEASED
+:      USER COULD STOP HERE AND PERFORM
+:      REMAINING COMMANDS LATER, OR HE
+:      MAY CONTINUE.
+:      DETERMINE INFLUENCE OF THIRD
+:      AIRCRAFT COMPONENT VIA ITERATION
+:      PROCEDURE.
+:  CNTL
NCMP=     1  ITERAT=      1
+:      REREAD ALL INFLUENCE SUB-MATRIX
+:      FILES.
+:  VNRE
FILE 7077,VN-PNC               HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7078,VN-PNC               HAS BEEN OPENED FOR READING ON UNIT   1
CREATION TIME=07/07/76     04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7079,VN-PNC               HAS BEEN OPENED FOR READING ON UNIT   1
```

FIGURE 8.7-2 Output for SOLN Code for Test Case No. 7 (Cont'd)

```
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7080,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7081,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7082,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7083,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7084,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
FILE 7085,VN-PNC            HAS BEEN OPENED FOR READING ON UNIT  1
CREATION TIME=07/07/76    04:09:01
UNIT  1 REWOUND AND RELEASED
+:      THE COMB COMMAND MAY NOT BE USED
+:      WITH THE ITERATION PROCEDURE.
+:      REREAD ALL CONSTANT SUB-VECTOR FILES.
+:  BCRE
FILE 7086,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7087,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:09:01
UNIT  2 REWOUND AND RELEASED
FILE 7088,BC-PNC            HAS BEEN OPENED FOR READING ON UNIT  2
CREATION TIME=07/07/76    04:09:01
UNIT  2 REWOUND AND RELEASED
+:      SOLVE DIAGONAL BLOCKS BY
+:      HOUSEHOLDER PROCEDURE, DEFINE THE
+:      SOLUTION SUB-VECTOR FILES, THE
+:      PRE-DETERMINED APPROXIMATE
+:      SOLUTION SUB-VECTOR FILES,
+:      MAXIMUM NUMBER OF ITERATIONS,
+:      AMOUNT OF OVER-RELAXATION,
+:      AND AMOUNT OF INTERMEDIATE
+:      PRINTOUT.
```

FIGURE 8.7-2 Output for SOLN Code for Test Case No. 7 (Cont'd)

```
+: SOLV
METHOD=        1
IPS=     2
IPRINT=      0
(OUTA,OUTANT,I=1,NCMP)=
 7093      0
 7094      0
 7095      0
(INPS,INPSNT,I=1,NCMP)=
 7091      0
 7092      0
    0      0
MAXIT1=     10
OMEGA=1.100
ITERP=       0

***  MATRIX INVERSE DEMANDED

ENTER SUB' HREC.       7 ROWS       5 COLUMNS.    MATRIX ON UNIT 11
EXIT HREC'  INVERSE ON UNIT 11

SUB. ATR TRANSPOSED A     7=BY=     5  MATRIX

***  MATRIX INVERSE DEMANDED

ENTER SUB' HREC.       6 ROWS       3 COLUMNS.    MATRIX ON UNIT 12
EXIT HREC'  INVERSE ON UNIT 12

SUB. ATR TRANSPOSED A     6=BY=     3  MATRIX

***  MATRIX INVERSE DEMANDED

ENTER SUB' HREC.       4 ROWS       2 COLUMNS.    MATRIX ON UNIT 11
EXIT HREC'  INVERSE ON UNIT 12

SUB. ATR TRANSPOSED A     4=BY=     2  MATRIX

SOLUTION VECTOR NO.  1
*********************
FILE 7086.BC=PNC           HAS BEEN OPENED FOR READING ON UNIT  4
```

FIGURE 8.7-2 Output for SOLN Code for Test Case No. 7 (Cont'd)

```
UNIT  4 REWOUND AND RELEASED
FILE 7087,BC-PNC              HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7088,BC-PNC              HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7091,SO-PNC              HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:16:43
UNIT  4 REWOUND AND RELEASED
FILE 7092,SO-PNC              HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:16:44
UNIT  4 REWOUND AND RELEASED

ITERATION     1  LARGEST RELATIVE ERROR=     .470115E+00

ITERATION     2  LARGEST RELATIVE ERROR=     .147611E+00

ITERATION     3  LARGEST RELATIVE ERROR=     .258860E-01

ITERATION     4  LARGEST RELATIVE ERROR=     .750495E-02

ITERATION     5  LARGEST RELATIVE ERROR=     .153576E-02

ITERATION     6  LARGEST RELATIVE ERROR=     .259851E-03

ITERATION     7  LARGEST RELATIVE ERROR=     .405204E-04
FILE 7093,SO-PNC/LIB$         HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:18:16
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   1  SOLUTIONS:
        .9999998E+00   .2000001E+01   .2999998E+01   .4000007E+01   .4999963E+01
FILE 7094,SO-PNC/LIB$         HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:18:18
UNIT  3 ENDFILED AND RELEASED

UNIT NO.  3
 COMPONENT NO.   2  SOLUTIONS:
        .6000016E+01   .6999996E+01   .8000001E+01
FILE 7095,SO-PNC/LIB$         HAS BEEN OPENED FOR WRITING ON UNIT  3
CREATION TIME = 07/22/76    11:18:19
UNIT  3 ENDFILED AND RELEASED
```

FIGURE 8.7-2 Output for SCLN Code for Test Case No. 7 (Cont'd)

```
UNIT NO,  3
 COMPONENT NO.   3  SOLUTIONS,
       .9000000E+01   .1000000F+02

NO. OF ITERATIONS TO CONVERGE=    7  LARGEST RELATIVE ERROR=    .405204E-04


SOLUTION VECTOR NO.  2
********************
FILE 7086,BC-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7087,BC-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7088,BC-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
UNIT  4 REWOUND AND RELEASED
FILE 7091,SO-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:16:43
UNIT  4 REWOUND AND RELEASED
FILE 7092,SO-PNC          HAS BEEN OPENED FOR READING ON UNIT  4
CREATION TIME=07/22/76    11:16:44
UNIT  4 REWOUND AND RELEASED

ITERATION     1  LARGEST RELATIVE ERROR=    .470115E+00

ITERATION     2  LARGEST RELATIVE ERROR=    .147611E+00

ITERATION     3  LARGEST RELATIVE ERROR=    .258860E-01

ITERATION     4  LARGEST RELATIVE ERROR=    .750495E-02

ITERATION     5  LARGEST RELATIVE ERROR=    .155576E-02

ITERATION     6  LARGEST RELATIVE ERROR=    .259851E-03

ITERATION     7  LARGEST RELATIVE ERROR=    .405204F-04
FILE 7093,SO-PNC/LIB$     HAS BEEN OPENED FOR WRITING ON UNIT  3
UNIT  3 ENDFILED AND RELEASED

UNIT NO,  3
 COMPONENT NO.   1  SOLUTIONS,
       .2000000E+01   .4000003E+01   .5999996E+01   .8000015F+01   .9999925E+01
```

FIGURE 8.7-2 Output for SCLN Code for Test Case No. 7 (Cont'd)

```
FILE 7094,SO-PNC/LIBS      HAS BEEN OPENED FOR WRITING ON UNIT   3
UNIT   3 ENDFILED AND RELEASED

UNIT NO,   3
 COMPONENT NO.    2  SOLUTIONS:
        .1200003E+02    .1399999E+02    .1600000E+02
FILE 7095,SO-PNC/LIBS      HAS BEEN OPENED FOR WRITING ON UNIT   3
UNIT   3 ENDFILED AND RELEASED

UNIT NO,   3
 COMPONENT NO.    3  SOLUTIONS:
        .1800000E+02    .2000000E+02

NO. OF ITERATIONS TO CONVERGE=    7  LARGEST RELATIVE ERROR=    .405204E-04
+!        ALL DONE.
+!   STOP
 STOP 777
```

FIGURE 8.7-2 Output for SCLN Code for Test Case No. 7 (Concluded)