371

YES

MSC INTERNAL TECHNICAL NOTE

MPF - A GENERALIZED PROGRAM FOR SOLVING NONLINEAR TWO-POINT BOUNDARY VALUE PROBLEMS

By

J. M. Lewallen
and
S. D. Williams

National Aeronautics and Space Administration

HOUSTON, TEXAS

FACILITY FORM 602

N70-76178 (ACCESSION NUMBER) (THRU)

186 (PAGES) none (CODE)

TMX 65626 (NASA CR OR TMX OR AD NUMBER) (CATEGORY)

Manned Spacecraft Center

MSC-ED-R-68-68

MSC INTERNAL TECHNICAL NOTE

MPF - A GENERALIZED PROGRAM FOR SOLVING NONLINEAR TWO-POINT BOUNDARY VALUE PROBLEMS

By

J. M. Lewallen

and

S. D. Williams

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas
June 1968

Revision date: November 1968 80006

MSC-ED-R-68-68

MPF - A GENERALIZED PROGRAM FOR SOLVING NONLINEAR TWO-POINT BOUNDARY VALUE PROBLEMS

Prepared by: J. M. Lewallen
Aerospace Technologist

Prepared by: S. D. Williams
Scientific Programming Specialist

Approved by: H. P. Decell, Jr.
Chief, Theory and Analysis Office

Approved by: Eugene H. Brock
Chief, Computational and Analysis Division

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

MPF - A GENERALIZED PROGRAM
FOR SOLVING NONLINEAR
TWO-POINT BOUNDARY VALUE PROBLEMS

Approved by: J. A. Barnes

J. A. Barnes
Supervisor, Theory and Analysis Section
Lockheed Electronics Company

Approved by: C. R. Wallace

C. R. Wallace
Manager, Scientific Analysis and
Programming Department
Lockheed Electronics Company

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

TABLE OF CONTENTS

	Page
SUMMARY	1
INTRODUCTION	2
THEORETICAL DEVELOPMENT	3
DESCRIPTION OF THE PROGRAM	10
MODIFICATIONS FOR DIFFERENT PROGRAM	26
DATA INPUT	32
EQUIPMENT	36
REFERENCES	37
APPENDICES	
A. EARTH-MARS TRANSFER EXAMPLE	A-1
1. FORMULATION	A-1
2. PROGRAM LISTING	A-5
3. FLOW CHART	A-68
4. SAMPLE INPUT	A-107
5. SAMPLE OUTPUT	B-109
B. BRACHISTOCHRONE EXAMPLE	B-1
1. FORMULATION	B-1
2. MODIFICATIONS FOR CUR SYSTEM LISTING	B-6
3. SAMPLE INPUT	B-8
4. SAMPLE OUTPUT	B-9

MPF - A GENERALIZED PROGRAM FOR SOLVING NONLINEAR TWO-POINT BOUNDARY VALUE PROBLEMS

By

J. M. Lewallen[1]

S. D. Williams[2]

SUMMARY

This report documents a numerical perturbation method that has had impressive success in solving the nonlinear two-point boundary value problem.

Many practical engineering problems are reduced mathematically to the nonlinear two-point boundary value problem. These problems arise quite naturally in areas of vibration, heat transfer, wave propagation, fluid mechanics, trajectory analysis and optimization theory. The method developed here requires the problem to be presented in terms of first-order, ordinary, differential equations with an adequate number of specified boundary conditions. The terminal conditions may be general functions of the problem variables. Several different convergence procedures are available for the user. With proper application, the chances are excellent that convergence may be achieved with only one computer run. If convergence has not been achieved, the information for a restart is provided automatically.

This routine is made available by the Theory and Analysis Office of the Computation and Analysis Division. The routine is generated in an effort to provide MSC engineers and contractor personnel with a proven method

[1] Manned Spacecraft Center, Houston, Texas
[2] Lockheed Electronics Company, Houston, Texas

for solving a popular class of mathematical problems. The purpose of this program is to reduce both the programming and computer time for the user, while maintaining the convenience of a general program. Office personnel will be available for consultation involving implementation of this program to the users problem. The authors would appreciate comments and suggestions that would improve the program.

INTRODUCTION

In analyzing scientific and engineering systems, it is often necessary to solve a nonlinear two-point boundary value problem. These problems arise naturally in the areas of vibration, heat transfer, wave propagation, fluid mechanics, trajectory analysis, and optimization theory. The equations that describe these dynamic systems are a set of first-order, nonlinear, ordinary differential equations (or may be put into this form). If n differential equations are involved, $n + 2$ boundary conditions must be specified. Assume that n initial values of the n dependent variables, the initial time, and the terminal time are given. In this case, straight forward numerical integration methods will yield a solution. If p $(p < n)$ initial dependent variables are specified at the initial time and q $(p + q = n)$ terminal dependent variables are specified at the terminal time, there is not a simple approach to the solution. Recourse is usually made to iterative methods.

Attempts to solve this split end-condition problem have been made for years, but until the advent of the high-speed digital computer most methods were considered impractical. In recent years, several efficient methods

have been implemented successfully. In 1963, Breakwell, Speyer and Bryson[1] proposed a method which considers trajectories perturbed from some reference path. These perturbed trajectories are generated when certain initial conditions are varied. After assuming the unknown initial conditions, each iteration produces corrections for the unknown initial conditions such that the terminal constraint error is reduced. Similar methods, considered here as perturbation methods, have been proposed by Jurovics and McIntyre[2] and Jazwinski.[3] In 1964, McGill and Kenneth[4] proposed the generalized Newton-Raphson method. Although this method has some distinct advantages over the perturbation methods, a detailed consideration of the method will not be made here. Similar methods have been proposed by Merriam[5], Sylvester and Meyer[6] and Roberts and Shipman[7]. An extensive comparison of some of the perturbation and quasilinearization methods has been made by Tapley and Lewallen[8] and several innovations to accelerate convergence rates have been proposed by Lewallen, Tapley and Williams[9].

THEORETICAL DEVELOPMENT

The problem is formulated in terms of a nonlinear, ordinary, first-order vector differential equation

$$\dot{z} = F(z,t) \tag{1}$$

where z is an n-vector and t is the independent variable time. If the initial time is specified, $n + 1$ boundary conditions are required for solution.

Suppose that p $(p < n)$ of the initial dependent variables are specified at the known initial time and that $q + 1$ $(p + q = n)$ relations

$$h(z_f, t_f) = 0 \tag{2}$$

are specified at the unknown terminal time. Since p of the initial variables are specified, it becomes convenient to arrange and partition the n-vector of initial conditions as

$$z^T(t_o) = \left[x^T(t_o) \;\vdots\; \lambda^T(t_o)\right] \tag{3}$$

where the vector $x(t_o)$ is composed of the p known initial conditions and the vector $\lambda(t_o)$ is composed of the q unknown initial conditions.

If Equation (1) is expanded in a Taylor's series about some reference trajectory, say the ith approximation to the correct solution, then

$$\dot{z}_{i+1} = \dot{z}_i + \left[\frac{\partial F}{\partial z}\right]_i \left[z_{i+1} - z_i\right] + \ldots \tag{4}$$

If the notation $\delta z = z_{i+1} - z_i$ is used, and only linear terms in the expression are retained, the linear perturbation equation is seen to be

$$\delta\dot{z} = \left[\frac{\partial F}{\partial z}\right] \delta z = A\delta z \tag{5}$$

where the matrix $A = \left[\partial F/\partial z\right]$ is evaluated on the ith trajectory.

In a similar manner, Equation (2) may be expanded about the terminal constraints associated with the ith trajectory. If terms higher than the first in dz_f and dt_f are neglected, the result may be expressed as

$$dh = \left[\frac{\partial h}{\partial z}\right]_f dz_f + \left[\frac{\partial h}{\partial t}\right]_f dt_f \tag{6}$$

Considering the effect of a variation in the terminal time leads to

$$dz_f = \delta z_f + \dot{z}_f dt_f \tag{7}$$

If this condition is substituted into Equation (6), the result becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \delta z_f + \dot{h}\, dt_f \tag{8}$$

where $\left[\partial h/\partial z\right]$ and $\dot{h}$ are evaluated at the terminal point for the ith approximation. If the terminal state variations can be expressed in terms of the initial state variations (i.e., $\delta z_f = \Pi\, \delta z_o$), Equation (8) becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \Pi\, \delta z_o + \dot{h}\, dt_f \tag{9}$$

where Π is an $n \times n$ matrix to be developed.

The method of perturbation functions (MPF) uses the perturbation equation, Equation (5), to generate a matrix of perturbation solutions. This is accomplished by integrating Equation (5) forward n times with the starting conditions

$$\tilde{\Phi}(t_o,t_o) = \left[\delta z_1(t_o), \ldots\ldots, \delta z_n(t_o)\right] = I \quad (10)$$

where I is the $n \times n$ unity matrix. The solution may be represented by

$$\delta z(t) = \tilde{\Phi}(t,t_o)\ \delta z(t_o) \quad (11)$$

where it is easily seen that $\Pi = \tilde{\Phi}$ and that Equation (9) becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \tilde{\Phi}(t_f,t_o)\ \delta z_o + \dot{h}\ dt_f \quad (12)$$

The problem may be simplified and the computational efficiency increased by partitioning the vector δz_o as shown in Equation (3). This requires partitioning the matrix $\tilde{\Phi}$ such that

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \left[\Psi \;\vdots\; \Phi\right] \begin{bmatrix}\delta x_o \\ \hline \delta\lambda_o\end{bmatrix} + \dot{h}\ dt_f \quad (13)$$

where Ψ is an $n \times p$ matrix and Φ is an $n \times q$ matrix. Since the p initial values of x_o are specified, $\delta x_o = 0$, then Equation (13) reduces to

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \Phi(t_f,t_o)\ \delta\lambda_o + \dot{h}\ dt_f \quad (14)$$

where

$$\Phi(t_o,t_o) = \begin{bmatrix} 0_{pxq} \\ \hline I_{qxq} \end{bmatrix} \tag{15}$$

The linear algebraic equation, Equation (14), represents $q + 1$ scalar equations in the $q + 1$ unknowns $\delta\lambda_o$ and dt_f ; i.e.,

$$y = C^{-1}dh \tag{16}$$

where $y^T = \left[\delta\lambda_o^T \,\vdots\, dt_f\right]$ and $C = \left\{\left[\frac{\partial h}{\partial z}\right] \Phi \,\vdots\, \dot{h}\right\}$. The computational efficiency is increased because the $\Phi(t_f,t_o)$ matrix requires only q vectors to be integrated rather than the n vectors required to generate $\Phi(t_f,t_o)$. The correction equation, Equation (16), requires the specification of a desired change in terminal error, dh . The philosophy associated with this correction is described in detail in Reference 9 and will be discussed briefly here.

The two major assumptions that have been made in this development are the linearization of both the differential equations and the terminal constraints. This linearization will not present severe limitations if successive trajectory iterations are near one another; i.e., the linearization assumptions are not stretched beyond the limits of validity. Hence, the <u>Normal Correction Procedure</u> of requesting that the change in terminal error, dh, for the next iteration equal the negative of the error, $-h$, on the last iteration may be too severe. In this case, divergence could occur.

To avoid this possibility, a Fractional Correction Procedure may be employed such that

$$dh = - ch \tag{17}$$

where $0 < c \leq 1.0$. Reference 9 also discusses the Minimum Norm Correction Procedure which allows each element of the h vector to have a different weight. Briefly, this procedure uses corrections obtained from either

$$y = -\left[C^T C + \beta I\right]^{-1} C^T h \tag{18}$$

or

$$y = -\left[C^T C + \alpha \operatorname{diag}(C^T C)\right]^{-1} C^T h \tag{19}$$

depending on which procedure is to be used. The β and α are parameters which describe the relation between gradient dominated corrections and Newton-Raphson dominated corrections. The Stepped-β and -α procedures use Equation (18) or (19) until the terminal norm is less than some prespecified value, γ . When this occurs, the Normal Correction Procedure is used to achieve convergence. When the Variable - β and - α procedures are being used, the parameters β and α are redefined on each iteration by

$$\beta = \beta_o \left[\frac{\|h\|^i}{\|h\|^{max}}\right]^{\rho} \tag{20}$$

and

$$\alpha = \alpha_o \left[\frac{\|h\|^i}{\|h\|^{max}} \right]^\rho \tag{21}$$

where β_o, α_o and ρ are chosen by the user.

The computation procedure for MPF is as follows:

(1) Arrange the n differential equations, Equation (1), such that the p equations having the specified initial conditions occur first.

(2) Integrate Equation (1) forward with the p known initial conditions and q assumed initial conditions. A terminal time, t_f, must be assumed to terminate the integration.

(3) Simultaneously with (2) above, Equation (5) must be integrated q times with the initial conditions shown in Equation (15). This results in the matrix $\Phi(t_f, t_o)$. The matrix $\left[\partial F/\partial z\right]$ is formed from the trajectory generated in (2).

(4) When the terminal time is reached, $\left[\partial h/\partial z\right]$ and $\dot{h}$ must be evaluated, and dh must be selected.

(5) The $q + 1$ algebraic equations, Equation (14), are solved for the $q + 1$ corrections $\delta\lambda_o$ and dt_f. One of the correction procedures must be chosen at this time.

(6) These corrections are applied to the assumed values of λ_o and t_f and the iterative procedure is repeated.

(7) Iterations are continued until either the corrections become sufficiently small or the terminal constraints are adequately satisfied.

DESCRIPTION OF THE PROGRAM

In this section, reference is made frequently to the variables introduced in the theoretical development section. To avoid ambiguity, program variables will be capitalized as will be subroutine names. For further clarity, parenthesis will be used to distinguish the program variable associated with the variable introduced in the theory.

MAIN

MAIN is the main program. It initializes data storage, and acts as a driver for various subprograms.

The program first transfers control to the subroutine (F5) that reads in the plotting labels.

The next step is to initialize the clock routine (RESET). The program then transfers control to the general input routine (F6) which reads all the information necessary to run a case. The initial data are saved to

restart from the same trial point with a different correction, if necessary (see CONVRG).

The card punch routine (CRDPCH) is called and the initial conditions of the n dependent variables, the beginning time, the final time, and the maximum norm of the terminal constraints are punched on cards compatible for a continuing run, if desired.

The clock routine (TIME) is called and the computational time is calculated in seconds. The initialization routine (F7) is called to establish the initial conditions for the differential equations and the perturbation equations. The print routine (FPRNT) is called twice; once for the printing of the clock time, and once for the printing of the initial conditions, if desired. The plot storage routine (FPLEW) is called to plot the initial conditions.

The integration routine (INTEG) is called and the n first-order differential equations are integrated forward. Simultaneously, the perturbation equations are integrated forward with the proper starting conditions. The print routine (FPRNT) and the plot storage routine (FPLEW) are called and the terminal conditions are printed and plotted, if desired. The clock routine (TIME) is called and the elasped time is calculated in seconds. The print routine is called to print the computational time required for the integration, if desired.

The convergence routine (CONVRG) and the plot routine (FPLOT) are called, and the desired graphs for the current iteration are plotted. If convergence has not been achieved, the next iteration is started. If convergence has been

achieved, the clock routine (TIME) is called and the total execution time is calculated in seconds. The print routine (FPRNT) is called to print the total time required for this case, if desired. The plot routine (FPLOT) is called and the desired graphs are generated. The routine will now attempt to execute another case.

Subroutine ABAM

This is a single step predictor-corrector technique used to solve a set of first-order differential equations. The method is based on an Adams-Bashford predictor (fourth-order) and an Adams-Moulton corrector (fifth-order) as discussed by Hildebrand.[10] Two iterations (METHOD =2) on the corrector will usually provide sufficient accuracy for most problems. This routine requires four back points and their derivatives.

The routine integrates, on any given iteration, the differential equations in F1 once, and the differential equations in F2 q times. Storage is facilitated by use of an index register (I1B and I2B) which allows integration to proceed from point to point by changing the index registers rather than moving the data. This gives a rolling drum effect.

The mathematical relations used are

$$Y_{n+1} = y_n + \frac{h}{24}\left(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}\right) + \frac{251}{720} h^5 y^{V}(\varepsilon_1)$$

for the predictor, and

$$Y_{n+1} = y_n + \frac{h}{720}\,(251Y'_{n+1} + 646y'_n - 264y_{n-1} + 106y'_{n-2} - 19y'_{n-3}) - \frac{3}{160}\,h^6 y^{vi}(\varepsilon_2)$$

for the corrector, where h is the integration step size.

Subroutine CONVRG

This routine determines the norm of the terminal constraints and implements the different correction schemes according to the value of KEY.

If KEY = 1, the Normal Correction Procedure is used. The change in the terminal error dh (B) is taken as the negative of the error in the terminal constraints h (HNEW), the coefficient matrix C (AA) is evaluated and the correction y (HNEW) is determined. The correction is added to the values on the last iteration (see ITER), and if the norm of the terminal constraints $\|h\|$ (HNN) and the individual corrections are less than some prescribed epsilon (EPS) convergence has been achieved. If convergence has not been achieved, the iteration count (ICOUNT) is checked against the maximum iterations (KK) allowed to determine if the iterative process is to be continued or terminated.

If KEY = 2, the Fractional Correction Procedure is used. The procedure is similiar to the case when KEY = 1 for determining a correction. However, the last two convergent solutions are preserved, and the fractional correction constant c (CI) is incremented by a small amount (DEL). If

divergence occurs, then the routine attempts to correct from the last convergent solution decrementing CI by DEL. When the correction vector is decremented to DEL and an improvement has not been made in the terminal error, the routine backs up to the second last convergent solution and tries again. If this does not achieve convergence, the case is aborted.

If KEY = 3, the Minimum Norm Correction Procedure is used. The routine first determines if either of the Stepped procedures are used (IFL $\leq$ 2) . In this case, the norm of the terminal constraints HNN is checked to determine if it is less than some prespecified value (FL2). If the norm is less than this value, control is switched to the Normal Correction Procedure. If the norm of the terminal constraints HNN is greater than FL2 or if the Variable procedures (IFL > 2) are used, the matrix C (AA) is formed (see F4). The matrix AA is printed in FPRNT, if desired, and modified in MLTPLY depending on which procedure is used. The modified matrix is then printed in FPRNT, if desired. The corrections are determined in CORVEC and printed, if desired. The routine then determines if convergence has been achieved as in KEY = 1.

Subroutine CORVEC

This subroutine forms the multiplication of a matrix, A , times a vector, B , and stores the answer in a vector, C . Mathematically, AB = C .

Subroutine CRDPCH

The purpose of this routine is to punch on cards the values of the dependent variable, initial and terminal time, and the maximum norm of the terminal constraints

for each iteration in the same format required for input.

If IPCH = 0, control is returned to the MAIN program. If IPCH $\neq$ 0, the N initial values of the dependent variables (DEPO) are punched under a 2D25.16 format. The initial time (TO) and the final time (TF) are punched with a 2D25.16 format. Finally, the maximum norm of the terminal constraints $\|h\|^{max}$ (HMAX) is punched under a D25.16 format.

Subroutine FPLEW

This routine stores the various parameters which are to be plotted on the high-speed drum. It uses the symbolic magnetic tape unit 3 to achieve this storage. The parameters which are stored, for the example shown in Appendix A, are the control angle versus time in days (STORA) and the norm of the terminal constraints versus the iteration count (STORB). For the convenience of the user, the control angle is calculated as a Y value, and time is calculated as an X value.

The program determines first, whether or not plotting is desired. If not (PLT $\leq$ 0), a return is executed. However, if plotting is desired (PLT > 0), the routine has four options dependent upon the argument of I (I = 5,7,9,11).

For I = 5, the plotting point index is initialized for the first point (K1 = 1). If this is the first iteration (ICOUNT = 1), the index is set to store the data in the first pair of columns (K2 = 1). It is then determined if this is the PLT^{th} (L) iteration. The initial number of points for this curve is set (INDEX(K2) = 1), and the data tape is

rewound. The extreme limits for the abscissa (XMAX, XMIN) and the ordinate (YMAX, YMIN) are set. The initial time in days (X(1)) and the control angle in degrees (Y(1)) are calculated. Control is transferred to the data manipulation portion of the routine.

For I = 7, the point and curve indexes are incremented by one (K1 = K1 + 1 and INDEX(K2) = INDEX(K2) + 1), the final time is computed in days (X(1)), and the final control angle is computed in degress (Y(1)). Control is transferred to the data manipulation portion of the routine.

For I = 9, it is determined if this is the IPLTth point (J = 0). For J $\neq$ 0, return to the integration routine (INTEG). If this point is to be plotted, the point index is incremented by one (K1 = K1 + 1), and the time in days and the control angle in degrees are calculated. Control is transferred to the data manipulation portion of the program.

For I = 11, the iteration count (STORB(ICOUNT,1)) and the norm of the terminal constraints (STORB(ICOUNT,2)) are saved, and control is returned to the convergence routine (CONVRG).

In the data manipulation portion of the routine, the actual extrema of the data are determined (XMAX, XMIN, YMAX, YMIN) and the trajectory points (X, Y) are stored into a data array (STORA) which will be placed on the high-speed drum (or magnetic tape unit 3). It is now determined if the data array (STORA) is full (K1 = NS) or if the last point has been calculated (K = 3). If neither of these conditions are met, control is returned to the calling routine (MAIN or INTEG).

If the data array is filled or this is the last point, the drum write counter (IWRT(K2)) is incremented, the maximum drum write count is evaluated (IWRT(1)), and the data is placed on the drum.

If this is the last point, the iteration count (IT2) is saved. If this is the first iteration (ICOUNT = 1), or if this is the PLT^{th} (L = 0) iteration, the curve index (K2) is incremented by one. A maximum of seven curves is saved (NS2 = 7).

If this is not the last point and this is not the first iteration, the data array (STORA) is filled with the previous information calculated for that block on previous trajectories, and the drum is backed up to allow new information to update the drum. The last point calculated is stored in the first row of the array. The plotting point index is reinitialized (K1 = 1), and the number of points for this curve is incremented by one (INDEX(K2)). Control is returned to the calling routine.

Subroutine FPLOT

This routine plots the information in STORA and STORB. The routine determines first if plots are desired (PLT > 0). If plots are not desired, control is returned to the main program.

If plots are desired, the number of curves (K2) that may occur on one grid is calculated from the iteration count (IT2), the curve plot frequency (PLT), and the maximum numbers of curves (NS2). If JPLT is zero, only the curves for

the current iteration are plotted; otherwise, all K2 curves are generated.

The plotting boundaries are determined, and the associated scaling factors for each curve are calculated. The curve data are read from the drum, and are plotted via the QUIKML routine. If the data were scaled, the scale factors are printed on the grid. If JPLT is zero, control is returned to the main program; otherwise, a plot illustrating the convergence history of this case is desired.

The routine now prepares for the semi-logarithmic plot of the norm of the terminal constraints versus the iteration count (STORB). The plot registers are cleared and the film is advanced one frame (RESET). Next the nonlinear mode is established for 8-cycle log in Y and linear in X (MODE). The labels for the Y axis are established, the span for both the X and Y axis is determined (SET1, SET2, XKK), and the grid increments (FIT) are calculated in rasters for the X axis. The grid routine (GRIDGN) is called and the curve is plotted (PLOT1). The X and Y axis identifiers are printed (D and C). The labels associated with the X axis (LABELX) and the Y axis (PRINT) are plotted.

Finally, the film is advanced (FILMAV), the buffers are dumped (DUMPBUF), and the linear mode is reestablished (MODE).

Subroutine FPRNT

This routine prints the intermediate information required during each iteration. Two printing modes are permitted, the suppressed mode (SWCH = 0) or the full mode (SWCH = 1).

Dependent upon the value (I) received from various routines, different information is printed. The information to be printed is established by the value of JSWCH (JSWCH = I + SWCH) through the use of a computed GO TO statement.

For I = 1 and SWCH = 1, the time in seconds is printed prior to the main program calling the integration routine. Control is returned to the main program. If SWCH = 0, there is no print action.

For I = 3 and SWCH = 1, the time in seconds is printed after calling the integration routine. Control is returned to the main program. If SWCH = 0 there is no print action.

For I = 5 and SWCH = 1, the initial time (TO) and the matrix representing the initial conditions for the differential equations and the preturbation equations (DEP) are printed. Control is returned to the main program. If SWCH = 0, there is no print action.

For I = 7 and SWCH = 1, the final time (VIND) and the matrix representing the final conditions for the differential equations and the perturbation equations (DEP) are printed. If SWCH = 0, the last q differential equations and final time are printed with the iteration number (ICOUNT). Control is returned to the main program.

For I = 9, the IPROth step causes the current time (VIND) and the matrix representing the differential equations and the perturbation equations (DEP) to be printed. Control is returned to the integration routine (INTRK5). The same action is caused by SWCH being either 0 or 1.

For I = 11 and SWCH = 1, the terminal constraint vector h (H) and the norm of the terminal constraints (HNN) are printed. If SWCH = 0, only the norm of the terminal constraints is printed. Control is returned to the convergence routine (CONVRG).

For I = 13 and SWCH = 1, the fractional correction constant (C1) is printed. If SWCH = 0, there is no print action. Control is returned to the convergence routine (CONVRG).

For I = 15 and SWCH = 1, the corrections y (H) that have been computed will be printed. If SWCH = 0, there is no print action. Control is returned to the convergence routine (CONVRG).

For I = 17 and SWCH = 1, the fractional correction constant (C1), the attempt number (ISET), the iteration number (ICOUNT), and the corrections (H) will be printed. If SWCH = 0, there is no print action. Control is returned to the convergence routine (CONVRG).

For I = 19 and SWCH = 1, the matrix A (A) is printed. If SWCH = 0, there is no print action. Control is returned to the convergence routine (CONVRG).

For I = 21 and SWCH = 1, the matrix computed by the Minimum Norm Correction Procedure (see MLTPLY) is printed. If SWCH = 0, there is no print action. Control is returned to the convergence routine (CONVRG).

For I = 23, the same action occurs as in I = 19.

For I = 25 and SWCH = 1, the total time in seconds required to complete the case is printed. If SWCH = 0, there is no print action. Control is returned to the main program.

Subroutine F1

This routine contains the n first-order differential equations that describe the dynamical system. The equations as presented in Equation (1) are

$$\dot{z} = F(z,t)$$

where $\dot{z}$ is represented in the program as P , z is represented as TY, and t is represented as T . The equations are arranged so that the equations having the p known initial conditions are first.

Subroutine F2

This routine contains the n perturbation equations. The equations as presented in Equation (5) are

$$\delta\dot{z} = A\delta z$$

where matrix A represents $[\partial F/\partial z]$ and is evaluated at each integration step.

In the first step, if the index JAY = 2, the procedure is initiated to evaluate the A matrix. Next, an index (NU) is checked to see if the midpoint for Runge-Kutta need be approximated. An index (J) is set depending on the calling routine or the stage of the Runge-Kutta routine. The coefficient matrix is set to zero, and the nonzero elements are

calculated. The product of the $[\partial F/\partial z]$ matrix (A) and δz (TY) is evaluated and stored into the derivative $\delta \dot{z}$ (P). The ordering of the equations in F2 is compatible with that in F1.

Subroutine F3

This routine evaluates the $q + 1$ terminal constraints h (H) from the terminal state $z(t_f)$ (DEP). This is achieved by satisfying a known terminal state which is input z_f (XF), i.e.,

$$h = z(t_f) - z_f$$

or by satisfying a functional relation dependent on the terminal values of the differential equations (DEP) and their derivatives (YPR), i.e.,

$$h = f(\dot{z},z,t)|_{t_f}$$

Subroutine F4

This routine evaluates the terminal error correction matrix $C = [(\partial h/\partial z)\Phi | \dot{h}]$ (A). The $(\partial h/\partial z)$ matrix (B) is evaluated at final time. The last column of A is set to $\dot{h}$ evaluated at final time. Finally, the matrix multiplication of $(\partial h/\partial z)$ (B) times Φ (DEP) is performed and stored in the first q columns of A.

Subroutine F5

This routine reads in two cards. The first card contains the plotting label for the X axis and the second card contains the plotting label for the Y axis.

Subroutine F6

This is the initial data input routine. Essentially all the data needed to run a case is read in this routine. In addition, most of the data read is printed to verify that the data were input correctly. The only information not read in this routine is that concerned with the plotting labels (See F5).

Subroutine F7

This routine sets up the initialization necessary to perform the integration of the n differential equations and the q n-vectors of initial conditions for the perturbation equations. The routine stores the n initial conditions for the differential equations (DEP0) into working storage (DEP and S), then the initial conditions for the perturbation equations are stored in the right-most portion of the remaining q columns of DEP.

Subroutine INTEG

This routine integrates the n-vector of differential equations and the q n-vectors of perturbation equations. The routine sets the step size for the Runge-Kutta integration and initializes the index registers to give a rolling drum effect. The Runge-Kutta technique yields the first four points and their derivatives to be used by the Adams-Predictor-Corrector technique. The Adams technique integrates forward until the final time (T2) is exceeded. Then the Runge-Kutta is called to integrate to the final time. Printing and plotting options are examined at each time step to see if this information is desired.

Subroutine INTRK5

This is a standard fourth-order Runge-Kutta integration technique (Reference 10). The storage is handled in the same manner as in ABAM. The mathematical formulation is

$$y_{n+1} = y_n + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3) + 0(h^5)$$

where

$$k_0 = h\ F(x_n, y_n)$$

$$k_1 = h\ F(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_0)$$

$$k_2 = h\ F(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_1)$$

$$k_3 = h\ F(x_n + h,\ y_n + k_2)$$

and h is the step size.

Subroutine ITER

This routine adds the corrections y (H) to the assumed final time (TF) and the q unknown initial λ's (DEPO). It is assumed that these are the last q elements of z (DEPO).

Subroutine MLTPLY

This routine sets up the matrix C (A) for the Minimum Norm Correction Procedure described in Reference 9.

The routine first evaluates C^TC (A^TA) and stores this into the temporary storage (B). The routine then checks each main diagonal element of B to guarantee that it is

larger than some prescribed value (BETA). If the element is not larger than this number, it is set to the largest main diagonal element. For IFL = 1, the Stepped-α procedure is used, i.e.,

$$B = \left[(A^T A + K) + \alpha \text{diag } (A^T A + K)\right]$$

For IFL = 2, the Stepped-β procedure is used, i.e.,

$$B = \left[(A^T A + K) + \beta I\right]$$

For IFL = 3, the Variable-α procedure is used, i.e.,

$$B = \left[(A^T A + K) + \alpha_0 \left(\frac{HNN}{HMAX}\right)^{\rho} \text{diagonal } (A^T A + K)\right]$$

For IFL = 4, the Variable-β procedure is used, i.e.,

$$B = \left[(A^T A + K) + \beta_0 \left(\frac{HNN}{HMAX}\right)^{\rho} I\right]$$

Finally, the routine transforms the error vector (C) into the modified error vector (D) and replaces the original matrix (A) by the newly computed matrix (B).

<u>Subroutine MINVDP</u>

See Reference 11.

MODIFICATIONS FOR DIFFERENT PROGRAM

This section is designed to assist the user in modifying the existing program to satisfy the users needs and requirements. To some extent, familiarity with the UNIVAC CUR system is useful and a basic knowledge of FORTRAN is assumed. All control cards are assumed to start in column 1 unless specified otherwise. An apostrophe in column 1 indicates a 7-8 multiple punch.

CONTROL CARD 1

$JOB card

CONTROL CARD 2

```
'N HDG       INPUT 10615
The message INPUT begins in column 13
```

CONTROL CARD 3

```
' ASG P=10615
```

CONTROL CARD 4

```
' XQT CUR
```

CONTROL CARDS 5-8 (all start in column 3)

```
TRW P
IN P
TRI P
TOC
```

CONTROL CARDS 9-10

```
'T FOR,* MAIN,MAIN,MAIN/B
-1,1
```

Insert the FORTRAN PARAMETER card equating N with the number of dependent variables, NQ with the q + 1 terminal constraints, KON with the number of special program constants required, and NS3 with the number of AXIS labels to be read in, i.e.,

```
PARAMETER  N = 8, NQ = 4, KON = 4, NS3 = 2
```

After this, the parameter card and the desired variables on the card will be indicated by ?. The user supplies the correct value for N, NQ, KON, and NS3.

CONTROL CARDS 11-12

```
'T FOR,* ABAM,ABAM,ABAM/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 13-14

```
'T FOR,* CONVRG,CONVRG,CONVRG/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 15-16

```
'T FOR,* CRDPCH,CRDPCH,CRDPCH/B
-2,2
    PARAMETER  N = ?
```

CONTROL CARDS 17-18

```
'T FOR,* FPLEW,FPLEW,FPLEW/B
-3,5'
    PARAMETER  N = ?, NQ = ?, NS3 = ?, KON = 4
```

DOUBLE PRECISION — anything required

CONTROL CARD 19

```
-61,62
    X(1) = information for the first X-axis
    Y(1) = information for the first Y-axis
    - - - - - - - - - - - - - - - - - - - -
    - - - - - - - - - - - - - - - - - - - -
    X(NS4) = information for the last X-axis
    Y(NS4) = information for the last Y-axis
    (Note:  NS4 = NS3/2, and is defined in a
    parameter statement.)
```

CONTROL CARDS 20-21

```
'T FOR,* FPLOT,FPLOT,FPLOT/B
-2,2
    PARAMETER  NS3 = ?
```

CONTROL CARDS 22-23

```
'T FOR,* FPRNT,FPRNT,FPRNT/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 24-25

```
'T FOR,* F1,F1,F1/B
-2,2
    PARAMETER  N = ?, KON = ?
```

CONTROL CARD 26

```
-5,6
    DOUBLE PRECISION local program constant names
    EQUIVALENCE (CON(1), local constant), ---,(CON(KON),
    local constant).
```

These two statements are only for those users that desire the use of an identifying label for their constants. They are not necessary for the proper use and execution of the program.

CONTROL CARD 27

-9,19

The set of N first-order differential equations P(I) which involves the dependent variables TY(I) with its corresponding coefficients.

CONTROL CARDS 28-29

```
'T FOR,* F2,F2,F2/B
-2,2
    PARAMETER  N = ?, KON = ?
```

CONTROL CARD 30

-5,7

Same comments under Control Card 26 apply here.

CONTROL CARD 31

-27,59

FORTRAN statements defining the N X N coefficient matrix of the perturbation equations. The matrix is A(I,J). The subscript I corresponds to the Ith equation and the J to the Jth coefficient in the equation. Only the nonzero coefficients need be evaluated.

CONTROL CARDS 32-33

```
'T FOR,* F3.F3,F3/B
-2,2
    PARAMETER  N = ?, NQ = ?, KON = ?
```

CONTROL CARD 34

-15,18

A set of FORTRAN statements evaluating the NQ elements of H(I) with the N dependent variables DEP(K,1), the N derivatives of the dependent variables YPR(J,I2B,1) and the NQ elements of the input terminal constraint XF(I).

CONTROL CARDS 35-36

```
'T FOR,* F4,F4,F4/B
-2,2
    PARAMETER  N = ?, NQ = ?, KON = ?
```

CONTROL CARD 37

```
-23,26
```

Place the nonzero elements of $[\partial h/\partial z]$ in the matrix B.

CONTROL CARD 38

```
-30,33
```

Place the NQ elements of $\dot{h}$ in A(I,NQ).

CONTROL CARDS 39-40

```
'T FOR,* F5,F5,F5/B
-2,2
    PARAMETER  NS3 = ?
```

CONTROL CARDS 41-42

```
'T FOR,* F6,F6,F6/B
-2,2
    PARAMETER  N = ?, NQ = ?, KON = ?
```

CONTROL CARDS 43-44

```
'T FOR,* F7,F7,F7/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 45-46

```
'T FOR,* INTEG,INTEG,INTEG/B
-2,2
    PARAMETER  N = ?
```

CONTROL CARDS 47-48

```
'T FOR,* INTRK5,INTRK5,INTRK5/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 49-50

```
'T FOR,* ITER,ITER,ITER/B
-2,2
    PARAMETER  N = ?, NQ = ?
```

CONTROL CARDS 51-52

```
'T FOR,* MLTPLY,MLTPLY,MLTPLY/B
-2,2
    PARAMETER  NQ = ?
```

CONTROL CARDS 53-55

```
' XQT CUR
      TOC (this starts in column 3)
' XQT MAIN/B
```

PROGRAM INPUT DATA

CONTROL CARD 56

```
' EOF
```

DATA INPUT

FORTRAN NOTATION	COLUMNS	FORMAT	DESCRIPTION
PLOT LABELS			
Card 1			
BCD(I,1), I=1,12	1-72	12A6	Plot label for the X-axis.
Card 2			
BCD(I,2), I=1,12	1-72	12A6	Plot label for the Y-axis.
DEPENDENT VARIABLES			
Cards 3-6			
DEPO(I),I=1,N	1-25 26-50	D25.16 D25.16	Initial values of the dependent variable. Input two to a card. (In this example N=8, see the PARAMETER card).
TERMINAL VALUES			
Cards 7-8			
XF(I),I=1,NQ	1-25 26-50	D25.16 D25.16	Desired terminal values. Input two to a card. (In this example NQ=4, see the PARAMETER card).
TIME INTERVAL			
Card 9			
TO	1-25	D25.16	Initial value of beginning integration time.
TF	26-50	D25.16	Initial value of final integration time.

FORTRAN NOTATION	COLUMNS	FORMAT	DESCRIPTION
ACCURACY CONTROL			
Card 10			
STEP	1-25	D25.16	Integration step size.
EPS	26-50	D25.16	Accuracy required of the terminal values.
RUN CONTROL			
Card 11			
METHOD	1-5	I5	Number of iterations with the Adams-Moulton corrector (always $\geq$ 1).
KK	6-10	I5	Maximum number of iterations allowed ($\leq$ 50).
IPLT	11-15	I5	Frequency of plotting points during the iteration. Require (TO-TF) < 475 (IPLT) STEP.
PLT	16-20	I5	Frequency of plotting iterations.
IPRO	21-25	I5	Frequency of printing points during each iteration.
KEY	26-30	I5	Correction procedure desired (KEY=1 is Normal Correction Procedure, KEY=2 is Fractional Correction Procedure, KEY=3 is Minimum Norm Correction Procedure).

FORTRAN NOTATION	COLUMNS	FORMAT	DESCRIPTION
SWCH	31-35	I5	Print Control Switch. (SWCH=0 is suppressed mode, SWCH=1 is full mode).
IPCH	36-40	I5	Punch Control Switch, (IPCH=0 no punching desired. IPCH=1, punch out the n dependent variables, the time interval, and HMAX on each iteration).
IHMAX	41-45	I5	Maximum Norm Control Switch. (IHMAX=0 do not read in HMAX, IHMAX≠0 read in HMAX).

CORRECTION SCHEME (not Used for KEY=1)

Card 12 (For KEY=2)

C	1-25	D25.16	Initial Fractional Correction Constant.
DEL	26-50	D25.16	Rate of change of Fractional Correction Constant.

ALTERNATE Card 12 (for KEY=3)

FL1	1-25	D25.16	α_o if IFL = 1 or 3 β_o if IFL = 2 or 4
FL2	26-50	D25.16	γ if IFL $\leq$ 2 ρ if IFL $\geq$ 3
IFL	55	I1	Denotes the Minimum Norm Correction Procedure to be used. (IFL=1 use the Stepped Alpha Procedure, IFL=2 use the Stepped Beta Procedure, IFL=3 use the Variable Alpha

FORTRAN NOTATION	COLUMNS	FORMAT	DESCRIPTION
			Procedure and IFL=4 use the Variable Beta Procedure).
MAXIMUM NORM (To be used only when IHMAX ≠ 0)			
Card 13 HMAX	1-25	D25.16	The maximum norm of the terminal constraints. (This makes sense only when used in conjunction with the Variable Alpha or Variable Beta Correction Procedures).
PROGRAM CONSTANTS			
Cards 14-17 PLACE	1-24	4A6	The alphanumeric identification the user identifies with the input constant.
CON(I)	25-50	D25.16	The program constant. (I is the subscript that ranges from 1 to KON. In this program KON=4, see PARAMETER card).

The next case starts at Card 3.

EQUIPMENT

This program has been checked out on the UNIVAC 1108 computer. In addition to the systems tapes, data input (5), data output (6), punch (-3), and SC4060 (17) tapes, two additional tapes are required. They are:

1. Program PCF tape (assigned P).
2. A scratch tape (3). This is usually unassigned, but when the drum is not sufficiently large to contain the plot data, it must be assigned (assigned 3) to a magnetic tape unit.

All input and output are consistent with the current requirements for handling by the UNIVAC 1108 peripheral equipment. The software requirement is a standard FORTRAN V system.

REFERENCES

1. Breakwell, J. V., Speyer, J. L, and Bryson, A. E., *Optimization and Control of Nonlinear Systems Using the Second Variation*, SIAM Journal on Control, Vol.1, No. 2, 1963.

2. Jurovics, S. A. and McIntyre, J. E., *The Adjoint Method and Its Application to Trajectory Optimization*, ARS Journal, Vol. 32, No. 9, 1962.

3. Jazwinski, A. H., *Optimal Trajectories and Linear Control of Nonlinear Systems*, AIAA Journal, Vol. 2, No. 8, 1964.

4. McGill, R. and Kenneth, P., *Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator*, AIAA Journal, Vol. 2, No. 10, 1964.

5. Merriam, C. W., *An Algorithm for the Iterative Solution of a Class of Two-Point Boundary Value Problems*, SIAM Journal on Control, Vol. 2, No. 1, 1964.

6. Sylvester, R. J. and Meyer, F., *Two Point-Boundary Problems by Quasilinearization*, SIAM Journal, Vol. 13, No. 2, 1965.

7. Roberts, S. M. and Shipman, J. S., *Continuation in Shooting Methods for Two-Point Boundary Value Problems*, Journal of Mathematical Analysis and Applications, Vol. 18, No. 1, 1967.

8. Tapley, B. D. and Lewallen, J. M., *Comparison of Several Numerical Optimization Methods*, Journal of Optimization Theory and Applications, Vol. 1, No. 1, 1967.

9. Lewallen, J. M., Tapley, B. D. and Williams, S. D., *Iteration Procedures for Indirect Trajectory Optimization Methods*, Journal of Spacecraft and Rockets, Vol. 5, No. 3, 1968.

10. Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill Book Company, 1956.

11. Clayton, E. G., *Compact Methods for Inverting Matrices and Solving Simultaneous Equations by Use of Gauss-Jordon Elimination*, Unpublished thesis submitted to the graduate school of Texas A&M for partial fulfillment of the requirements for the degree of Master of Science.

Appendix A

EARTH-MARS TRANSFER EXAMPLE

1. FORMULATION

The minimum time Earth-Mars transfer problem may be stated as follows: Determine the control history $\psi(t)$ such that a spacecraft may transfer from initial conditions corresponding to that of Earth to conditions corresponding to that of Mars in minimum time.

The differential equations of motion are

$$\dot{x}_1 = \dot{u} = v^2/r - GM/r^2 + (T/m) \sin \psi$$

$$\dot{x}_2 = \dot{v} = - uv/r + (T/m) \cos \psi$$

$$\dot{x}_3 = \dot{r} = u$$

$$\dot{x}_4 = \dot{\theta} = v/r$$

where u, v, r, and θ are the radial velocity, tangential velocity, radial position and angular position, respectively. The control variable ψ is the thrust orientation to the local horizontal. The symbol GM is the gravitational constant of the sun, and $m = m_o - \dot{m}t$ is the vehicle mass.

When the optimization process is applied, the control variable angle ψ is eliminated from the above equations and four additional equations are required to be satisfied (Euler-Lagrange equations).

The initial boundary conditions $\mu_3(t_o) = -1$ is used in place of one of the terminal transversality boundary conditions and upon reordering, the differential equations for F1 become

$$\dot{z}_1 = \dot{x}_1 = \dot{u} = v^2/r - GM/r^2 - (T/m)\ \mu_1/\sqrt{\mu_1^2 + \mu_2^2}$$

$$\dot{z}_2 = \dot{x}_2 = \dot{v} = -\ uv/r - (T/m)\ \mu_2/\sqrt{\mu_1^2 + \mu_2^2}$$

$$\dot{z}_3 = \dot{x}_3 = \dot{r} = u$$

$$\dot{z}_4 = \dot{x}_4 = \dot{\theta} = v/r$$

$$\dot{z}_5 = \dot{x}_5 = \dot{\mu}_3 = (v^2/r^2 - 2GM/r^3)\ \mu_1 - (uv/r^2)\ \mu_2 + (v/r^2)\ \mu_4$$

$$\dot{z}_6 = \dot{\lambda}_1 = \dot{\mu}_4 = 0$$

$$\dot{z}_7 = \dot{\lambda}_2 = \dot{\mu}_1 = (v/r)\ \mu_2 - \mu_3$$

$$\dot{z}_8 = \dot{\lambda}_3 = \dot{\mu}_2 = -\ (2v/r)\ \mu_1 + (u/r)\ \mu_2 - (1/r)\ \mu_4$$

where $t = t_o$		$t = t_f$ (unspecified)	
$u(t_o)$	= 0.0	$u(t_f)$	= 0.0
$v(t_o)$	= 1.0	$v(t_f)$	= 0.8078
$r(t_o)$	= 1.0	$r(t_f)$	= 1.532
$\theta(t_o)$	= 0.0	$\mu_4(t_f)$	= 0.0
$\mu_3(t_o)$	= - 1.0		

The perturbation equations for F2 are

$$\delta\dot{z}_1 = (2v/r)\ \delta z_2 + (2GM/r^3 - v^2/r^2)\ \delta z_3$$

$$- \left[T\mu_2/m(\mu_1^2 + \mu_2^2)^{3/2}\right] \left[\mu_2 \delta z_7 - \mu_1 \delta z_8\right]$$

$$\delta\dot{z}_2 = -(v/r)\ \delta z_1 - (u/r)\ \delta z_2 + (uv/r^2)\ \delta z_3$$

$$+ \left[T\mu_1/m(\mu_1^2 + \mu_2^2)^{3/2}\right] \left[\mu_2 \delta z_7 - \mu_1 \delta z_8\right]$$

$$\delta\dot{z}_3 = \delta z_1$$

$$\delta\dot{z}_4 = (1/r)\ \delta z_2 - (v/r^2)\ \delta z_3$$

$$\delta\dot{z}_5 = -(v\mu_2/r^2)\ \delta z_1 + \left[(2v\mu_1 - u\mu_2 + \mu_4)/r^2\right]\ \delta z_2$$

$$+ \left[(6GM\mu_1/r - 2v^2\mu_1 + 2uv\mu_2 - 2v\mu_4)/r^3\right]\ \delta z_3$$

$$+ (v/r^2)\ \delta z_6 + (v^2/r^2 - 2GM/r^3)\ \delta z_7$$

$$- (uv/r^2)\ \delta z_8$$

$$\delta\dot{z}_6 = 0$$

$$\delta\dot{z}_7 = (\mu_2/r)\ \delta z_2 - (v\mu_2/r^2)\ \delta z_3 - \delta z_5 + (v/r)\ \delta z_8$$

$$\delta\dot{z}_8 = (\mu_2/r)\ \delta z_1 - (2\mu_1/r)\ \delta z_2$$

$$+ \left[(2v\mu_1 - u\mu_2 + \mu_4)/r^2\right]\ \delta z_3 - (1/r)\ \delta z_6$$

$$- (2\ v/r)\ \delta z_7 + (u/r)\ \delta z_8$$

The terminal constraints for F3 are

$$
\begin{aligned}
h_1 &= u(t_f) \\
h_2 &= v(t_f) - 0.8078 \\
h_3 &= r(t_f) - 1.532 \\
h_4 &= \mu_4(t_f)
\end{aligned}
$$

The partial derivative of the terminal constraints with respect to the dependent variables and the time rates of change of the terminal constraints for F4 are

$$
\left[\frac{\partial h}{\partial z}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}
$$

and

$$
\dot{h} = \begin{bmatrix} \dot{u}\ (t_f) \\ \dot{v}\ (t_f) \\ \dot{r}\ (t_f) \\ \dot{\mu}_4(t_f) \end{bmatrix}
$$

2. PROGRAM LISTING

```
PARAMETER N=6, NQ=4, KON=4, NS3=2
PARAMETER NS=50, NS1=50, NS2=7, NS4=NS3/2
COMMON/BC1/BACK(N),BACK2(N),HICK(NQ),HICK2(NQ),CIOLD,CIOLD2,TFB,
1          TFB2,ICOUNT,KOUNT
DOUBLE PRECISION BACK,BACK2,HICK,HICK2,CIOLD,CIOLD2,TFB,TFB2
COMMON/TIMER/XJ,XMEX,XN
COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
DOUBLE PRECISION DEP,YPR
COMMON/INITAL/DEPO(N),TO,TF,EPS
DOUBLE PRECISION DEPO,TO,TF,EPS
COMMON/PRNT/SWCH
INTEGER SWCH
COMMON/EVAL/VIND,DELT,INDI
DOUBLE PRECISION VIND,DELT
COMMON/MIDTR/J,JKK
COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
INTEGER PLT
DOUBLE PRECISION STEP
COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
DOUBLE PRECISION A,H,XF
COMMON/LEWALN/HNN,HMAX,NTHRU
DOUBLE PRECISION HNN,HMAX
COMMON/NEWTON/CI,CRAT
DOUBLE PRECISION CI,CRAT
COMMON/TRL/B(NQ),ISET
DOUBLE PRECISION B
COMMON/CONST/CON(KON)
```

```
      DOUBLE PRECISION CON
      COMMON/DIFEQ/P(N),TY(N),T
      DOUBLE PRECISION  P,TY,T
      COMMON/MODE/S(N,3),JAY,NU,IONE,ITWO,IERRCT
      DOUBLE PRECISION S
      COMMON/COUNT/METHOD,ILOC,I1B,I2B,I3B,K2B
      COMMON/FLAG/FL1,FL2,IFL
      DOUBLE PRECISION FL1,FL2
      COMMON/KPCH1/IPCH,IHMAX
      COMMON/LOGIC/BOOL
      LOGICAL BOOL
      COMMON/STUFF/XMAX(NS4,NS2),YMAX(NS4,NS2),XMIN(NS4,NS2),IT2,JPLT,
     1             YMIN(NS4,NS2),STORB(NS1,2),INDEX(NS2),IWRT(NS2)
      COMMON/PLOTO/BCD(12,NS3)
      CALL F5
    3 CONTINUE
      CALL RESET
                XMEX = 0.0
                NTHRU = 1
      CALL F6
      DO 403 J=1,N
      BACK(J)=DEPO(J)
  403 CONTINUE
      IFB=IF
      KOUNT=0
      CIOLD=CI
                ICOUNT = 0
    3           ICOUNT = ICOUNT + 1
      CALL CRDPCH
```

```
            CALL TIME(J)
            XJ = J
            XJ = XJ/1000.0
            XK = XJ
            XMEX = XJ - XMEX
      CALL F7
      IT=1
      CALL FPRNT(IT)
      IT=5
      CALL FPRNT(IT)
      CALL FPLEW(IT)
      CALL INTEG
      IT=7
      CALL FPRNT(IT)
      CALL FPLEW(IT)
            CALL TIME(J)
            XJ = J
            XJ = XJ/1000.0
            XN = XJ - XK
            XMEX=XJ
      IT=3
      CALL FPRNT(IT)
      CALL CONVRG(KIK)
            JPLT = 0
            CALL FPLOT
      GO TO (5,35),KIK
35          CALL TIME(J)
            XJ = J
            XJ = XJ/1000.0
```

```
      IT=25
      CALL FPRNT(IT)
                JPLT = 1
      CALL FPLOT
40              CONTINUE
                GO TO 5
       END
```

```
      SUBROUTINE ABAM
      PARAMETER N=8, NQ=4
      COMMON/EVAL/VIND,DELT,INDI
      DOUBLE PRECISION VIND,DELT
      COMMON/COUNT/METHOD,ILOC,I1B,I2B,I3B,K2B
      COMMON/DIFEQ/P(N),DEPVAR(N),XN
      DOUBLE PRECISION P,DEPVAR,XN
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION DEP,YPR
      COMMON/MODE/S(N,3),JAY,NU,IONE,ITWO,IERRCT
      DOUBLE PRECISION S
      DOUBLE PRECISION XIP(N)
      DOUBLE PRECISION A(5),B(4)
      DATA (B(J),J=1,4)/-.375D0,.1541666666666666D1,-.2458333333333333D1
     1,.2291666666666666D1/
      DATA (A(J),J=1,5)/-.2638888888888888D-1,.1472222222222222D0,
     1-0.3666666666666667D0, .8972222222222222D0, .3486111111111111D0/
C
C     THIS ROUTINE INTEGRATES THE DIFFERENTIAL EQUATIONS USING AN
C
C     ADAMS-BASHFORD PREDICTOR WITH AN ADAMS-MOULTON CORRECTOR.
C
      NS=NQ
            VIND = VIND + DELT
      XN=VIND
            DO 40 JAY = 1,NS
               ISAV = METHOD
```

```
              DO 10  J = 1,N
                 DEPVAR(J) = DEP(J,JAY) + DELT*(B(1)*YPR(J,I3B,JAY)
  1              + B(2)*YPR(J,I2B,JAY) + B(3)*YPR(J,I1B,JAY) +
  1              B(4)*YPR(J,ILOC,JAY))
10               XIP(J) = DEP(J,JAY) + DELT*(A(1)*YPR(J,I3B,JAY) +
  1              A(2)*YPR(J,I2B,JAY) + A(3)*YPR(J,I1B,JAY) +
  1              A(4)*YPR(J,ILOC,JAY))
   IF(JAY.NE.1) GO TO 12
   CALL F1
   GO TO 14
12 CONTINUE
   CALL F2
14 CONTINUE
15            DO 20 J = 1,N
                 DEP(J,JAY) = DELT*A(5)*P(J) + XIP(J)
20               DEPVAR(J) = DEP(J,JAY)
   IF(JAY.NE.1) GO TO 22
   CALL F1
   GO TO 24
22 CONTINUE
   CALL F2
24 CONTINUE
              IF(ISAV.EQ.1) GO TO 25
              ISAV = ISAV - 1
              GO TO 15
25            DO  30 J = 1,N
30               YPR(J,I3B ,JAY) = P(J)
              IF(JAY.NE.1) GO TO 40
              DO 35 J = 1,N
```

```
35                S(J,ITWO) = DEPVAR(J)
40             CONTINUE
            ISAV = I3B
            I3B = I2B
            I2B = I1B
            I1B = ILOC
            ILOC = ISAV
            ISAV = IONE
            IONE = ITWO
            ITWO = ISAV
      RETURN
      END
```

```
      SUBROUTINE CONVRG(KIK)
      PARAMETER N=6, NQ=4
      COMMON/MATRIX/AA(NQ,NQ),HNEW(NQ),XF(NQ)
      DOUBLE PRECISION AA,HNEW,XF
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION DEP,YPR
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
      COMMON/TRL/B(NQ),ISET
      DOUBLE PRECISION B
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
      COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
      INTEGER PLT
      DOUBLE PRECISION STEP
      COMMON/FLAG/FL1,FL2,IFL
      DOUBLE PRECISION FL1,FL2
      COMMON/BC1/BACK(N),BACK2(N),HICK(NQ),HICK2(NQ),CIOLD,CIOLD2,TFB,
     1           TFB2,ICOUNT,KOUNT
      DOUBLE PRECISION BACK,BACK2,HICK,HICK2,CIOLD,CIOLD2,TFB,TFB2
      COMMON/NEWTON/CI,CRAT
      DOUBLE PRECISION CI,CRAT
      COMMON/KPCH1/IPCH,IHMAX
      DOUBLE PRECISION HNO,HNO2,XN,X(NQ)
      DIMENSION J2(NQ),I2(NQ)
      LOGICAL BOOL
      KIK=1
```

```
C
C       DETERMINE THE TERMINAL NORM
C
      CALL F3
C
C       CALCULATE THE NORM OF THE TERMINAL CONSTRAINTS
C
              HNN=0.D0
      DO 15 J=1,NQ
   15               HNN = HNN + HNEW(J)*HNEW(J)
              HNN=DSQRT(HNN)
      ITT=11
      CALL FPRNT(ITT)
      CALL FPLEW(ITT)
C
C       EVALUATE THE MAXIMUM NORM OF THE TERMINAL CONSTRAINTS
C
      IF(IHMAX.NE.0)GO TO 50
      IF(NTHRU.EQ.1) HMAX=HNN
   50 CONTINUE
      HMAX=DMAX1(HMAX,HNN)
      GO TO(100,200,300),KEY
  100 CONTINUE
C
C       NORMAL CORRECTION PROCEDURE
C
C       EVALUATE THE MATRIX AND SET UP THE DISSATISFACTION VECTOR
C
       CALL F4
```

```
      DO 130 J=1,NQ
           B(J)=-HNEW(J)
  130 CONTINUE
C
C      EVALUATE THE INVERSE AND ADD ON THE CORRECTIONS
C
  150 CONTINUE
      CALL MINVDP(AA,NQ,K,X,J2,I2)
      IF(K.EQ.1) GO TO 1500
      CALL CORVEC(AA,B,NQ,HNEW)
      ITT=15
      CALL FPRNT(ITT)
      CALL ITER
C
C     DETERMINE IF CONVERGENCE HAS BEEN ACHEIVED
C
  160 BOOL = .TRUE.
      DO 180 J=1,NQ
         XN=DABS(HNEW(J))
         IF(XN.GT.EPS) BOOL = .FALSE.
  180 CONTINUE
      IF(HNN.LE.EPS .AND. BOOL) GO TO 70
      NTHRU=2
      GO TO 5
  200 CONTINUE
C
C      FRACTIONAL PROCEDURE
C
      IF(NTHRU.EQ.1) GO TO 220
```

```
      IF(HNN.GE.HNO) GO TO 250
      DO 210 J=1,N
              BACK2(J)=BACK(J)
              BACK(J)=DEPO(J)
  210 CONTINUE
      TFB2=TFB
      TFB=TF
C
C     DETERMINE THE FRACTIONAL CORRECTION
C
      CI=CI+CRAT
      IF(CI.GT.1.D0)CI=1.D0
C
C     EVALUATE THE MATRIX AND SET UP THE DISSATISFACTION VECTOR
C
  220 CALL F4
      ISET=1
      ISET1=0
      IF(NTHRU.EQ.1) ISET1=1
      DO 230 J=1,NQ
              B(J)=-CI*HNEW(J)
  230 CONTINUE
      ITT=19
      CALL FPRNT(ITT)
C
C     EVALUATE THE INVERSE AND ADD ON THE CORRECTIONS
C
      CALL MINVDP(AA,NQ,K,X,J2,I2)
      IF(K.EQ.1) GO TO 1500
```

```
      CALL CORVEC(AA,B,NQ,HNEW)
      ITT=17
      CALL FPRNT(ITT)
      CALL ITER
      KOUNT=KOUNT+1
      DO 240 J=1,NQ
              HICK2(J)=HICK(J)
              HICK(J)=HNEW(J)
  240 CONTINUE
      CIOLD2=CIOLD
      CIOLD=CI
      HNO2=HNO
      HNO=HNN
      GO TO 160
  250 IF(CI.EQ.CRAT) GO TO 280
  255 ISET=ISET+1
      ICOUNT=ICOUNT-1
      KOUNT=KOUNT+1
      DO 260  I=1,N
              DEPO(I)=BACK(I)
  260 CONTINUE
      IF(ISET.LE.2) CI = CIOLD
      TF=TFB
      CI=CI-CRAT
      IF(CI.LT.CRAT) CI=CRAT
      DO 270 J=1,NQ
              HNEW(J)=CI*HICK(J)/CIOLD
  270 CONTINUE
      ITT=17
```

```
      CALL FPRNT(ITT)
      CALL ITER
      GO TO 5
  280 IF(ISET1.EQ.1) GO TO 1000
      IF(ICOUNT.LT.2) GO TO 1000
      ISET1=1
      DO 285 I=1,N
              BACK(I)=BACK2(I)
  285 CONTINUE
      DO 290 J=1,NQ
              HICK(J)= HICK2(J)
  290 CONTINUE
      CIOLD=CIOLD2
      TFB=TFB2
      HNO=HNO2
      ISET=0
      ICOUNT=ICOUNT-1
      GO TO 255
  300 CONTINUE
  310 IF(IFL.GT.2) GO TO 320
      IF(HNN.LT.FL2) KEY=1
      IF(KEY.EQ.1) GO TO 100
C
C      FORM THE MATRIX AND SET UP THE DISSATISFACTION VECTOR
C
  320 CALL F4
      ITT=19
      CALL FPRNT(ITT)
      CALL MLTPLY
```

```
      ITT=21
      CALL FPRNT(ITT)
      HNO=HNN
      CALL MINVDP(AA,NQ,K,X,D2,I2)
      IF(K.EQ.1) GO TO 1500
      CALL CORVEC(AA,B,NQ,HNEW)
      CALL ITER
      ITT=15
      CALL FPRNT(ITT)
      GO TO 160
   70 WRITE(6,71)
   71 FORMAT(1H0,19X,66H**************** CONVERGENCE HAS BEEN ACHEIVED
     1****************** )
       GO TO 35
 1000 WRITE(6,1010)
 1010 FORMAT(1H0,19X,66H*************** CONVERGENCE HAS NOT BEEN ACHEIVE
     1D **************** )
      GO TO 35
 1500 CONTINUE
      WRITE(6,1510)
 1510 FORMAT(1H0,41X,22HTHE MATRIX IS SINGULAR  )
      GO TO 1000
    5 CONTINUE
      IF(TF.LT. 0.D0) GO TO 1000
      IF(ICOUNT.EQ.KK) GO TO 1000
      IF(KOUNT.EQ.KK) GO TO 1000
      RETURN
   35 CONTINUE
      KIK=2
```

```
RETURN

END
```

```
      SUBROUTINE CORVEC(A,B,NQ,C)
      DOUBLE PRECISION A(NQ,NQ),B(NQ),C(NQ)
C
C     THIS ROUTINE TAKES THE MATRIX A MULTIPLIES IT TIMES THE VECTOR B
C
C     AND STORES THE RESULT IN THE VECTOR C.
C
      DO 10 I=1,NQ
            C(I)=0.D0
            DO 5 J=1,NQ
                  C(I)=C(I)+A(I,J)*B(J)
    5       CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CRDPCH
      PARAMETER N=8
      PARAMETER IUNIT=-3
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
      COMMON/KPCH1/IPCH,IHMAX
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
  100 FORMAT(2D25.16)
      IF(IPCH.EQ.0)RETURN
C
C     IF IPCH = 0  RETURN.  FOR IPCH NE 0  THE INITIAL VALUES OF THE
C
C     DEPENDENT VARIABLES, BEGINNING AND FINAL TIME, AND THE MAXIMUM
C
C     NORM OF THE TERMINAL CONSTRIINTS IS PUNCHED OUT ON CARDS.
C
      WRITE(IUNIT,100)DEPO
      WRITE(IUNIT,100)TO,TF
      WRITE(IUNIT,100)HMAX
      RETURN
      END
```

```
      SUBROUTINE FPLEW(I)
      PARAMETER N=8, NQ=4, NS3=2, KON=4
      DOUBLE PRECISION TIME,DEG
      DATA TIME/.58132256D2/,DEG/.57295779513082320D2/
      PARAMETER NS=50, NS1=50, NS2=7, NS4=NS3/2
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      COMMON/STUFF/XMAX(NS4,NS2),YMAX(NS4,NS2),XMIN(NS4,NS2),IT2,JPLT,
     1             YMIN(NS4,NS2),STORB(NS1,2),INDEX(NS2),IWRT(NS2)
C
C     INFORMATION IS STORED COLUMN-WISE
C
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION DEP,YPR
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
      COMMON/EVAL/VIND,DELT,INDI
      DOUBLE PRECISION VIND,DELT
      COMMON/MIDTR/J,JKK
      COMMON/BC1/BACK(N),BACK2(N),HICK(NQ),HICK2(NQ),CIOLD,CIOLD2,TFB,
     1           TFB2,ICOUNT,KOUNT
      DOUBLE PRECISION BACK,BACK2,HICK,HICK2,CIOLD,CIOLD2,TFB,TFB2
      COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
      DOUBLE PRECISION STEP
      INTEGER PLT
```

```
      COMMON/COUNT/METHOD,KLOC,K1B,K2B,K3B,I2B
      DIMENSION X(NS4),Y(NS4)
      DATA A1/.1E37/
      DIMENSION STORA(NS,NS3,NS2)
C
C     IPLT IS THE FREQUENCY OF PLOTTING EACH POINT PER CURVE
C
C     PLT  IS THE FREQUENCY OF PLOTTING EACH ITERATION IF 0 OR LESS
C          NO PLOTTING IS DESIRED
C
      IF(PLT.LE.0)RETURN
      K=I-4
      GO TO (1,100,2,100,3,100,4),K
    1 K1=1
      IF(ICOUNT.EQ.1) K2=1
      L=MOD(ICOUNT,PLT)
      INDEX(K2) = 1
      IWRT(K2) = 0
      REWIND 3
      IF(ICOUNT.EQ.1) GO TO 15
      READ(3) STORA
      REWIND 3
   15 CONTINUE
      DO 20 IR=1,NS4
                XMAX(IR,K2) = -A1
                YMAX(IR,K2) = -A1
                XMIN(IR,K2) = A1
                YMIN(IR,K2) = A1
   20 CONTINUE
```

```
      GO TO 5
    2 K1=K1+1
      INDEX(K2) = INDEX(K2) + 1
    5 CONTINUE
      X(1) = TIME*VINU
      Y(1) = DEG*DATAN2(DEP(7,1),DEP(8,1))
      GO TO 100
    6 CONTINUE
      IT2=ICOUNT
      IF(ICOUNT.EQ.1.OR. L.EQ.0) K2=K2+1
      IF(K2.GT.NS2) K2=NS2
      RETURN
    4 STORB(ICOUNT,1)=ICOUNT
      STORB(ICOUNT,2)=HNN
      RETURN
    3 IF(J.EQ.0) GO TO 2
      RETURN
  100 CONTINUE
      DO 120 LL=1,NS4
                LJ = LL + LL
                LK = LJ - 1
                STORA(K1,LK,K2) = X(LL)
                STORA(K1,LJ,K2) = Y(LL)
                XMAX(LL,K2) = AMAX1(XMAX(LL,K2),X(LL))
                YMAX(LL,K2) = AMAX1(YMAX(LL,K2),Y(LL))
                XMIN(LL,K2) = AMIN1(XMIN(LL,K2),X(LL))
                YMIN(LL,K2) = AMIN1(YMIN(LL,K2),Y(LL))
  120 CONTINUE
      IF(K1.LT.NS  .AND.  K.NE.3) RETURN
```

```
            IWRT(K2) = IWRT(K2) + 1
            IWRT(1) = MAX0(IWRT(1),IWRT(K2))
            WRITE(3)STORA
      IF(K.EQ.3) GO TO 6
      IF(ICOUNT.EQ.1) GO TO 130
      READ(3) STORA
      BACKSPACE 3
  130 CONTINUE
      DO 140 LL=1,NS4
            LJ = LL + LL
            LK = LJ - 1
            STORA(1,LK,K2) = X(LL)
            STORA(1,LJ,K2) = Y(LL)
  140 CONTINUE
            K1 = 1
            INDEX(K2) = INDEX(K2) + 1
      RETURN
      END
```

```
      SUBROUTINE FPLOT
      PARAMETER NS3=2
      PARAMETER NS=50, NS1=50, NS2=7, NS4=NS3/2
      COMMON/STUFF/XMAX(NS4,NS2),YMAX(NS4,NS2),XMIN(NS4,NS2),IT2,JPLT,
     1             YMIN(NS4,NS2),STORB(NS1,2),INDEX(NS2),IWRT(NS2)
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
      COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
      DOUBLE PRECISION STEP
      INTEGER PLT
      COMMON/PLOTO/BCD(12,NS3)
      DIMENSION A(32),B(24),C(6),D(3),X(NS),Y(NS),ISYMB(25),G(2),WO(2)
      DIMENSION STORA(NS,NS3,NS2)
      DATA FIT2/112.5/
      DATA (A(I),I=1,24)/1.E+14,1.E+13,1.E+12,1.E+11,1.E+10,1.E+09,
     1                   1.E+08,1.E+07,1.E+06,1.E+05,1.E+04,1.E+03,
     2                   1.E+02,1.E+01,1.E+00,1.E-01,1.E-02,1.E-03,
     3                   1.E-04,1.E-05,1.E-06,1.E-07,1.E-08,1.E-09/
C
C
      DATA (B(I),I=1,32)/6H1.E+14, 6H1.E+13, 6H1.E+12, 6H1.E+11,
     1                   6H1.E+10, 6H1.E+09, 6H1.E+08, 6H1.E+07,
     2                   6H1.E+06, 6H1.E+05, 6H1.E+04, 6H1.E+03,
     3                   6H1.E+02, 6H1.E+01, 6H1.E+00, 6H1.E-01,
     4                   6H1.E-02, 6H1.E-03, 6H1.E-04, 6H1.E-05,
     5                   6H1.E-06, 6H1.E-07, 6H1.E-08, 6H1.E-09,
     6                   6H1.E-10, 6H1.E-11, 6H1.E-12, 6H1.E-13,
```

```
     7                    6H1.E-14, 6H1.E-15, 6H1.E-16, 6H1.E-17/
C
C
      DATA (C(I),I=1,06)/6HNORM O, 6HF THE , 6HTERMIN, 6HAL CON,
     1                    6HSIRAIN, 6HTS    /
C
C
      DATA (D(I),I=1,03)/6HITERAT, 6HION CO, 6HUNT   /
C
C
      DATA E1/1.0E+06/,E2/1.0E-01/,(G(I),I=1,2)/6HSCALE ,6HFACTOR/
C
C
      DATA (ISYMB(I),I=1,25)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HI
     1,1HC,1HD,1HE,1HF,1HG,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ/
C
C
C
C     DETERMINE IF PLOTTING IS DESIRED.
C
      IF(PLT.LE.0)RETURN
C
C     DETERMINE THE MAXIMUM NUMBER OF CURVES TO BE PLOTTED (K2).
C
      LD = IT2 + 2 * ( PLT - 1 )
      LD = LD / PLT
      K2 = MIN0(LD,NS2)
      KT = 1
      IF(JPLT.EQ.0) KT = K2
```

```
      IWRT1 = IWRT(KT)
C
C     IF JPLT IS ZERO -- ONLY THE CURVE FOR THE CURRENT ITERATION IS
C     DESIRED, OTHERWISE PLOT ALL K2 CURVES.
C
      DO 200 I=1,NS3,2
               I1 = (I+1)/2
               LP = -1
               REWIND 3
C
C     DETERMINE THE PLOTTING BOUNDARIES.
C
               XR = XMAX(I1,KT)
               XL = XMIN(I1,KT)
               YT = YMAX(I1,KT)
               YB = YMIN(I1,KT)
               IF(KT.EQ.K2) GO TO 115
               DO 110 K=2,K2
                        XR = AMAX1(XR,XMAX(I1,K))
                        XL = AMIN1(XL,XMIN(I1,K))
                        YT = AMAX1(YT,YMAX(I1,K))
                        YB = AMIN1(YB,YMIN(I1,K))
  110          CONTINUE
  115          CONTINUE
C
C     DETERMINE THE SCALE FACTORS FOR X AND Y SUCH THAT X,Y LT 1.E+06.
C
               SX = 1.0
               SY = 1.0
```

```
              XS = AMAX1(ABS(XR),ABS(XL))
              YS = AMAX1(ABS(YT),ABS(YB))
 120          CONTINUE
                       IF(XS.LT.E1) GO TO 125
                       SX = SX*E2
                       XS = XS*E2
              GO TO 120
 125          CONTINUE
                       IF(YS.LT.E1) GO TO 130
                       SY = SY*E2
                       YS = YS*E2
              GO TO 125
 130          CONTINUE
C
C     SCALE THE PLOT BOUNDARIES
C
              XR = XR*SX
              XL = XL*SX
              YT = YT*SY
              YB = YB*SY
              DO 150 J=1,IWRT1
                       JP = NS*(J-1)
C
C     READ THE DATA TAPE
C
                       READ(3) STORA
                       DO 140 K=KT,K2
C
C     SCALE THE K-TH CURVE FOR THIS GRID
```

```
C
                                        NP = INDEX(K) - JP
                                        IF(NP.LT.1) GO TO 140
                                        NP = MIN0(NP,NS)
                                        ISYM = ISYMB(K)
                                        DO 135 L=1,NP
                                                X(L) = STORA(L,I,K)*SX
                                                Y(L) = STORA(L,1+I,K)*ST
  135                                   CONTINUE
C
C     PLOT ALL DATA ON THE SAME GRID AND IF MORE DATA MUST BE READ FROM
C          TAPE 3 ** PLOT IT ALSO
C
                                        CALL QUIKML(LP,XL,XR,YB,YT,ISYM,
     1                                          BCD(1,I),BCD(1,1+I),NP,X,Y)
                                        LP = 0
  140                       CONTINUE
  150             CONTINUE
C
C     IF THE VARIABLES WERE SCALED PRINT THE SCALE FACTORS ON THE GRID
C
                  IF(SX.GT..5) GO TO 160
                  CALL BINDEC(SX,NC,WO(1),WO(2))
                  CALL PRINT(474, 5,8,0,12,G)
                  CALL PRINT(578, 5,8,0,NC,WO)
  160             CONTINUE
                  IF(SY.GT..5) GO TO 170
                  CALL BINDEC(SY,NC,WO(1),WO(2))
                  CALL PRINT(474,20,8,0,12,G)
```

```
              CALL PRINT(578,20,8,0,NC,WO)
  170         CONTINUE
C
C     PLOT THE REMAINING GRIDS
C
  200 CONTINUE
C
C     DETERMINE IF THIS IS THE END OF A JOB
C
      REWIND 3
      IF(JPLT.EQ.0) RETURN
C
C
C     PREPARE FOR SIMI-LOG GRIDS
C
C
      CALL RSET(0)
      CALL MODE(0,1,8,8)
      K1=0
      DO 5 I=1,24
      IF(HMAX.GE.A(I)) GO TO 10
      K1=I
    5 CONTINUE
   10 K2=K1
      IF(K2.EQ.0)K2=1
      SET2=A(K2)
      SET1=SET2*1.E-08
      XKK=FLOAT(KK)
      IF(XKK.LT.10.) XKK = 10.
```

```
      FIT=900./XKK
      FIX=50.
      IT1=-IT2
      CALL GRIDGN(100,1000,050,0950,FIT,FIT,5,5)
      CALL PLOT1(1,1,0.,XKK,SET1,SET2,STORB(1,1),STORB(1,2),IT1,1,1HX)
      CALL PRINT(486,980,8,0,15,D)
      CALL PRINT(20,244,0,16,32,C)
      JK=KK+1
      DO 20 JI=1,JK
      I=JI-1
      CALL LABELX(I,1,0)
   20 CONTINUE
      K1=K2-1
      DO 30 I=1,9
      IFIX=FIX
      CALL PRINT(50,IFIX,8,0,6,B(I+K1))
      FIX=FIX+FIT2
   30 CONTINUE
      CALL FILMAV(0)
      CALL DMPBUF
      CALL MODE(0,0,8,8)
      RETURN
      END
```

```
SUBROUTINE FPRNT(I)
PARAMETER N=8 , NQ=4
COMMON/COUNT/METHOD,KLOC,K1B,K2B,K3B,I2B
COMMON/TIMER/XJ,XMEX,XN
COMMON/BC1/BACK(N,2),BACK2(NQ,2),BACK3(4),ICOUNT,KOUNT
DOUBLE PRECISION BACK,BACK2,BACK3
COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
DOUBLE PRECISION DEP,YPR
COMMON/INITAL/DEPO(N),TO,TF,EPS
DOUBLE PRECISION DEPO,TO,TF,EPS
COMMON/PRNT/SWCH
INTEGER SWCH
COMMON/EVAL/VIND,DELT,INDI
DOUBLE PRECISION VIND,DELT
COMMON/MIDTR/J,JKK
COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
INTEGER PLT
DOUBLE PRECISION STEP
COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
DOUBLE PRECISION A,H,XF
COMMON/LEWALN/HNN,HMAX,NTHRU
DOUBLE PRECISION HNN,HMAX
COMMON/NEWTON/CI,CRAT
DOUBLE PRECISION CI,CRAT
COMMON/TRL/B(NQ ),ISET
DOUBLE PRECISION B
JSWCH=I+SWCH
```

```
      GO TO(90,100,90,120,90,130,160,140,150,150,175,170,90,180,90,190,
     X      90,200,90,210,90,220, 90,190,90,110),JSWCH
   90 RETURN
  100 WRITE(6,101)ICOUNT,XJ,XMEX
  101 FORMAT(1H1,37X,14HBEGINNING THE ,I3,13H TH ITERATION ,/,21X,
     1        51HTIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS  ,
     2        F12.4,/,38X,17HELAPSED TIME WAS  ,F12.4)
      GO TO 90
  110 WRITE(6,111)XJ
  111 FORMAT(1H0,37X,18HTHE FINAL TIME IS ,F12.4)
      GO TO 90
  120 WRITE(6,121)XJ,XN
  121 FORMAT(1H0,23X,45HTIME AT COMPLETION OF FORWARD INTEGRATION IS ,
     1        F12.4,/,38X,17HELAPSED TIME WAS  ,F12.4)
      GO TO 90
  130 WRITE(6,131) TO
  131 FORMAT(1H0,22X,34HTHE DEPENDENT VARIABLES AT TIME =  ,D25.16)
  135 DO 139 K2=1,NQ,2
            K3=K2+1
            IF(K3.GT.NQ) GO TO 137
            WRITE(6,136)(K1,K2,DEP(K1,K2),K1,K3,DEP(K1,K3),K1=1,N)
  136 FORMAT( //,(20X,I2,1X,I2,1X,D25.16,4X,I2,1X,I2,1X,D25.16))
            GO TO 139
  137       WRITE(6,138)(K1,K2,DEP(K1,K2),K1=1,N)
  138 FORMAT( //,(37X,I2,1X,I2,2X,D25.16))
  139 CONTINUE
      GO TO 90
  140 WRITE(6,131)VIND
      GO TO 135
```

```
150 IF(J.NE.0) GO TO 90
    WRITE(6,151)JKK,VIND
151 FORMAT(1H0,24X,14HPRINT COUNT = ,I5,5X,7HTIME = ,D25.16)
    GO TO 135
160 NQ1=NQ-1
    NP=N-NQ1
    WRITE(6,161)ICOUNT,NQ1
161 FORMAT(1H0,44X,I2,13H TH ITERATION,/,25X,35HDISPLAYING FINAL TIME
   1AND THE LAST  ,I2,20H DEPENDENT VARIABLES )
    WRITE(6,162)TF,(K1,DEPV(NP+K1),K1=1,NQ1)
162 FORMAT(1H0,37X,5HTF = ,D25.16,/,(21X,I2,2X,D25.16,6X,I2,2X,D25.16)
   1)
    GO TO 90
170 WRITE(6,171)(K,H(K),K=1,NQ)
171 FORMAT(1H0,37X,30HTHE TERMINAL CONSTRAINT VECTOR ,//,(21X,I2,2X,
   1         D25.16,6X,I2,2X,D25.16))
175 WRITE(6,176)HNN
176 FORMAT(/,22X,37HTHE NORM OF THE TERMINAL CONSTRAINTS  ,D25.16)
    GO TO 90
180 WRITE(6,181)CI
181 FORMAT(/,23X,35HTHE FRACTIONAL CORRECTION CONSTANT  ,D25.16)
    GO TO 90
190 WRITE(6,191)ICOUNT,(K,H(K),K=1,NQ)
191 FORMAT(/,33X,23HTHE CORRECTIONS AT THE ,I2,13H TH ITERATION ,//,
   1         (21X,I2,2X,D25.16,6X,I2,2X,D25.16))
    GO TO 90
200 WRITE(6,181)CI
    WRITE(6,201)ISET,ICOUNT
201 FORMAT(1H0,22X,12HTHIS IS THE ,I2,32H TH ATTEMPT TO CORRECT FROM T
```

```
   1HE ,I2,13H TH ITERATION )
    GO TO 190
210 CONTINUE
    WRITE(6,225)
    DO 219 K2=1,NQ,2
            K3=K2+1
            IF(K3.GT.NQ) GO TO 217
            WRITE(6,136)(K1,K2,A(K1,K2),K1,K3,A(K1,K3),K1=1,NQ)
            GO TO 219
217         WRITE(6,138)(K1,K2,A(K1,K2),K1=1,NQ)
219 CONTINUE
    GO TO 90
220 CONTINUE
    WRITE(6,230)
    DO 222 K2=1,NQ,2
            K3=K2+1
            IF(K3.GT.NQ) GO TO 221
            WRITE(6,136)(K1,K2,A(K1,K2),K1,K3,A(K1,K3),K1=1,NQ)
            GO TO 222
221         WRITE(6,138)(K1,K2,A(K1,K2),K1=1,NQ)
222 CONTINUE
225 FORMAT(1H0,47X,12HTHE A MATRIX  )
230 FORMAT(1H0,46X,14HTHE A*A MATRIX  )
    GO TO 90
    END
```

```
      SUBROUTINE F1
      PARAMETER N=8, KON=4
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      DOUBLE PRECISION  WOGD,WOED,TH,GM
      EQUIVALENCE (CON(1),GM),(CON(2),WOGD),(CON(3),WOED),(CON(4),TH)
      COMMON/DIFEQ/P(N),TY(N),T
      DOUBLE PRECISION  P,TY,T
      DOUBLE PRECISION  XNN ,DSQRT
      XNN=(WOGD-WOED*T)*DSQRT(TY(7)**2+TY(8)**2)
      P(1)=TY(2)**2/TY(3)-GM/ TY(3)**2-TH*TY(7)/XNN
      P(2)=-TY(1)*TY(2)/TY(3)-TH*TY(8)/XNN
      P(3)=TY(1)
      P(4)=TY(2)/TY(3)
      P(5)=(((TY(2)**2-2.D0*GM/TY(3))*TY(7))-(TY(1)*TY(2)*TY(8))+(TY(2)*
     1              TY(6)))/(TY(3)**2)
      P(6)=0.D0
      P(7)=TY(2)*TY(8)/TY(3)-TY(5)
      P(8)=(-2.D0*TY(2)*TY(7)+TY(1)*TY(8)-TY(6))/TY(3)
      RETURN
      END
```

```
      SUBROUTINE F2
      PARAMETER N=8, KON=4
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      DOUBLE PRECISION  WOGD,WOED,TH,GM
      EQUIVALENCE (CON(1),GM),(CON(2),WOGD),(CON(3),WOED),(CON(4),TH)
      DOUBLE PRECISION R,R2,R3,DSQRT
      COMMON/DIFEQ/P(N),TY(N),T
      DOUBLE PRECISION  P,TY,T
      COMMON/MODE/S(N,3),JAY,NU,IONE,ITWO,IERRCT
      DOUBLE PRECISION  S
      DOUBLE PRECISION A(N,N)
      IF(JAY.NE.2) GO TO 30
      IF(NU.NE.1) GO TO 15
      DO 10 J=1,N
              S(J,3)=.5D0*(S(J,IONE)+S(J,ITWO))
   10 CONTINUE
   15 CONTINUE
      J=3
      IF(NU.EQ.1)J=IONE
      IF(NU.EQ.4)J=ITWO
      DO 25 I=1,N
      DO 20 K=1,N
      A(I,K)=0.D0
   20 CONTINUE
   25 CONTINUE
      R2=S(7,J)**2+S(8,J)**2
```

```
      R=(WOGD-WOED*T)*DSQRT(R2)
      R3=TH/(R*R2)
      R=S(3,J)
      A(1,2)=(2.D0*S(2,J))/R
      A(1,3)=(2.D0*GM/R-S(2,J)**2)/R**2
      A(1,7)=-R3*S(8,J)**2
      A(1,8)=R3*S(7,J)*S(8,J)
      A(2,1)=-S(2,J)/R
      A(2,2)=-S(1,J)/R
      A(2,3)=(S(1,J)*S(2,J))/R**2
      A(2,7)=R3*S(7,J)*S(8,J)
      A(2,8)=-R3*S(7,J)**2
      A(3,1)=1.D0
      A(4,2)=1.D0/R
      A(4,3)=-S(2,J)/R**2
      A(5,1)=-(S(2,J)*S(8,J))/R**2
      A(5,2)=(2.D0*S(2,J)*S(7,J)-S(1,J)*S(8,J)+S(6,J))/R**2
      A(5,3)=((6.D0*GM/R-2.D0*S(2,J)**2)*S(7,J)
     1       +2.D0*S(2,J)*(S(1,J)*S(8,J)-S(6,J)))/R**3
      A(5,6)=S(2,J)/R**2
      A(5,7)=(S(2,J)**2-2.D0*GM/R)/R**2
      A(5,8)=-(S(1,J)*S(2,J))/R**2
      A(7,2)=S(8,J)/R
      A(7,3)=-(S(2,J)*S(8,J))/R**2
      A(7,5)=-1.D0
      A(7,8)=S(2,J)/R
      A(8,1)=S(8,J)/R
      A(8,2)=-(2.D0*S(7,J))/R
      A(8,3)=(2.D0*S(2,J)*S(7,J)-S(1,J)*S(8,J)+S(6,J))/R**2
```

```
      A(8,6)=-1.D0/R
      A(8,7)=-(2.D0*S(2,J))/R
      A(8,8)=S(1,J)/R
   30 CONTINUE
      DO 40 I=1,N
      P(I)=0.D0
      DO 35 J=1,N
      P(I)=P(I)+A(I,J)*TY(J)
   35 CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE F3
      PARAMETER N=8, NQ=4, KON=4
      PARAMETER NQ1=NQ-1
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION A,H,XF,DEP,YPR
      COMMON/COUNT/METHOD,KLOC,K1B,K2B,K3B,I2B
C
C     THIS ROUTINE DETERMINES THE DISSATISFACTION IN THE TERMINAL
C
C     VALUES OF THE DEPENDENT VARIABLE.
C
      DO 10 J=1,NQ1
              H(J)=DEP(J,1)-XF(J)
   10 CONTINUE
              H(NQ)=DEP(6,1)-XF(NQ)
       RETURN
       END
```

```
      SUBROUTINE F4
      PARAMETER N=8, NQ=4, KON=4
      PARAMETER NQ1=NQ-1
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION A,H,XF,DEP,YPR
      COMMON/COUNT/METHOD,KLOC,K1B,K2B,K3B,I2B
      DOUBLE PRECISION B(NQ,N)
C
C     YPR(J,I2B,1) CONTAINS Z-DOT.  THE J CORRESPONDS TO THE I-TH H-DOT.
C
C     NOW PLACE THE NON-ZERO ELEMENTS OF PARTIAL OF H WITH RESPECT TO Z
C
C     IN B.
C
      DO 20 J=1,N
      DO 10 I=1,NQ
      B(I,J)=0.D0
   10 CONTINUE
   20 CONTINUE
      B(1,1)=1.D0
      B(2,2)=1.D0
      B(3,3)=1.D0
      B(4,6)=1.D0
C
```

```
C     PLACE THE NQ ELEMENTS OF H-DOT IN A(I,NQ).
C
      A(1,NQ)=YPR(1,I2B,1)
      A(2,NQ)=YPR(2,I2B,1)
      A(3,NQ)=YPR(3,I2B,1)
      A(4,NQ)=YPR(6,I2B,1)
      DO 60 I=1,NQ
      DO 50 J=1,NQ1
      A(I,J)=0.D0
      L=J+1
      DO 40 K=1,N
      A(I,J)=A(I,J)+B(I,K)*DEP(K,L)
   40 CONTINUE
   50 CONTINUE
   60 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE F5
      PARAMETER NS3=2
      COMMON/PLOTO/BCD(12,NS3)
    2  FORMAT(12A6)
C
C     READ IN THE LABELS FOR THE X AND Y AXES.
C
      READ(5,2)BCD
       RETURN
       END
```

```
      SUBROUTINE F6
      PARAMETER N=8, NQ=4, KON=4
      COMMON/CONST/CON(KON)
      DOUBLE PRECISION CON
      DIMENSION PLACE(4)
      COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
      DOUBLE PRECISION A,H,XF
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
      COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
      DOUBLE PRECISION STEP
      COMMON/COUNT/METHOD,I1B,I2B,I3B
      COMMON/NEWTON/C,DEL
      DOUBLE PRECISION C,DEL
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
      COMMON/KPCH1/IPCH,IHMAX
      COMMON/FLAG/FL1,FL2,IFL
      DOUBLE PRECISION FL1,FL2
      COMMON/PRNT/SWCH
      INTEGER SWCH,PLT
   10 FORMAT(1H1,36X,32HMETHOD OF PERTURBATION FUNCTIONS ,//,
     1            37X,32HTWO-POINT BOUNDARY VALUE PROBLEM ,//)
   20 FORMAT(1H0,32X,40HINITIAL VALUE OF THE DEPENDENT VARIABLES ,//)
   25 FORMAT(20X,I3,3X,D25.16,4X,I3,3X,D25.16)
   30 FORMAT(1H0,32X,40HDESIRED VALUES OF THE TERMINAL VARIABLES  ,//)
   40 FORMAT(1H0,39X,25HTHE INITIAL TIME INTERVAL ,/,20X,6HFROM   ,
```

```
     1          D25.16,10H  /   TO   ,D25.16)
  50 FORMAT(1H0,25X,29HTHE INTEGRATION STEP SIZE IS  ,D25.16)
  60 FORMAT(1H0,19X,61HTHE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON
     1CORRECTOR IS  ,I5)
  80 FORMAT(1H0,27X,44HTHE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS ,I5)
  90 FORMAT(1H0,19X,41HTHE ACCURACY REQUIRED FOR TERMINATION IS  ,
     1          D25.16)
 100 FORMAT(1H0,22X,5HEVERY,I5,33HTH POINT WILL BE PLOTTED FOR EACH ,
     1          I5,12HTH ITERATION )
 110 FORMAT(1H0,26X,5HEVERY,I5,42HTH POINT WILL BE PRINTED ON EACH ITER
     1ATION )
 120 FORMAT(1H0,39X,23HPRINT CONTROL SWITCH =  ,I2)
 125 FORMAT(1H0,39X,23HPUNCH CONTROL SWITCH =  ,I2)
 130 FORMAT(1H0,26X,I2,50H SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICA
     1TIONS ,//)
 135 FORMAT(23X,4A6,11X,D25.16)
 140 FORMAT(1H0,38X,27HNORMAL CORRECTION PROCEDURE )
 150 FORMAT(1H0,37X,31HFRACTIONAL CORRECTION PROCEDURE ,//,
     1          20X,4HC =  ,D25.16,6X,6HDEL =  ,D25.16)
 160 FORMAT(1H0,36X,33HMINIMUM NORM CORRECTION PROCEDURE )
 161 FORMAT(1H0,40X,23HSTEPPED ALPHA PROCEDURE ,/,20X,7HALPHA =,D25.16,
     1          9H  GAMMA = ,D25.16)
 162 FORMAT(1H0,41X,22HSTEPPED BETA PROCEDURE ,/,20X,7HBETA = ,D25.16,
     1          9H  GAMMA = ,D25.16)
 163 FORMAT(1H0,40X,24HVARIABLE ALPHA PROCEDURE ,/,20X,7HALPHA = ,
     1          D25.16,6X,3HP = ,D25.16)
 164 FORMAT(1H0,40X,23HVARIABLE BETA PROCEDURE ,/,20X,7HBETA = ,D25.16,
     1          5X,4HP = ,D25.16)
 165 FORMAT(1H1)
```

```
  170 FORMAT(1H0,23X,34HTHE MAXIMUM NORM IS ASSUMED TO BE  ,D25.16)
  210 FORMAT(2D25.16)
  220 FORMAT(14I5)
  230 FORMAT(4A6,1X,D25.16)
  240 FORMAT(2D25.16,4X,I1)
C
C     THE FORMAT STATEMENTS UNDER 200 ARE FOR OUTPUT
C
C     THE FORMAT STATEMENTS OVER  200 ARE FOR INPUT
C
      WRITE(6,10)
      WRITE(6,20)
      READ(5,210)DEPO
      WRITE(6,25)(J,DEPO(J),J=1,N)
      WRITE(6,30)
      READ(5,210)XF
      WRITE(6,25)(J,XF(J),J=1,NQ)
      READ(5,210)TO,TF
      WRITE(6,40)TO,TF
      READ(5,210)STEP,EPS
      WRITE(6,50)STEP
      READ(5,220)METHOD,KK,IPLT,PLT,IPRO,KEY,SWCH,IPCH,IHMAX
      IF(SWCH.NE.0) SWCH=1
      WRITE(6,80)KK
      WRITE(6,90)EPS
      WRITE(6,60)METHOD
      WRITE(6,100)IPLT,PLT
      WRITE(6,110)IPRO
      WRITE(6,120)SWCH
```

```
      WRITE(6,125)IPCH
      GO TO(310,320,330),KEY
310   WRITE(6,140)
      GO TO 340
320   READ(5,210)C,DEL
      WRITE(6,150)C,DEL
      GO TO 340
330   WRITE(6,160)
      READ(5,240)FL1,FL2,IFL
      GO TO(331,332,333,334),IFL
331   WRITE(6,161)FL1,FL2
      GO TO 340
332   WRITE(6,162)FL1,FL2
      GO TO 340
333   WRITE(6,163)FL1,FL2
      GO TO 340
334   WRITE(6,164)FL1,FL2
340   CONTINUE
      IF(IHMAX.EQ.0) GO TO 345
      READ(5,210)HMAX
      WRITE(6,170)HMAX
345   CONTINUE
      NOK=KON
      IF(NOK.LE.0)GO TO 400
      WRITE(6,130)NOK
      DO 350 I=1,KON
      READ(5,230)PLACE,CON(I)
      WRITE(6,135)PLACE,CON(I)
350   CONTINUE
```

```
400 IF(SWCH.EQ.0)WRITE(6,165)
    RETURN
    END
```

```
      SUBROUTINE F7
      PARAMETER N=8 , NQ=4
      PARAMETER NP=N-NQ
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION DEP,YPR
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
      COMMON/MODE/S(N,3),INN(5)
      DOUBLE PRECISION S
      COMMON/EVAL/VIND,DELT,INDI
      DOUBLE PRECISION VIND,DELT
C
C     SET UP THE INITIAL CONDITIONS FOR INTEGRATION
C
      VIND=TO
 1002          DO 25 J = 1,N
                  DEP(J,1) = DEPO(J)
                  S(J,1) = DEPO(J)
                  DO 25 JAY=2,NQ
               DEP(J,JAY) = 0.0D0
   25          CONTINUE
              DO 30 I=2,NQ
                    J=NP+I
                    DEP(J,I)=1.0D0
   30         CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTEG
      PARAMETER N=8
      COMMON/MIDTR/J,JKK
      COMMON/NUM/STEP,KEY,KK,PLT,IPLT,IPRO
      DOUBLE PRECISION STEP
      INTEGER PLT
      COMMON/COUNT/METHOD,ILOC,I1B,I2B,I3B,K2B
      COMMON/LOGIC/BOOL
      LOGICAL BOOL
      COMMON/EVAL/VIND,DELT,IND1
      DOUBLE PRECISION VIND,DELT
      COMMON/MODE/S(N,3),JAY,NU,IONE,ITWO,IERRCT
      DOUBLE PRECISION S
      COMMON/INITAL/DEPO(N),T1,T2,EPS
      DOUBLE PRECISION DEPO,T1,T2,EPS
      DOUBLE PRECISION XN,ERR
      DATA ERR/3.D0/,IERR/3/,JERR/9/
C
C     MAIN INTEGRATION ROUTINE.  THIS ROUTINE USES A RUNGE-KUTTA STARTER
C
C     AND AN ADAMS PREDICTOR-CORRECTOR.  THE LAST STEP IS ALSO DONE WITH
C
C     A RUNGE-KUTTA.
C
            DELT = STEP/ERR
            BOOL = .FALSE.
            IKEY = 1
```

```
            INDI = 1
            I48 = 1
            IONE = 1
            ITWO = 2
            IERRCT = 1
            ILOC = 1
            I1B = 2
            I2B = 3
            I3B = 4
            VIND = T1
            LTHL = JERR
 5          DO 35 JKK = I48,LTHL
            K2B = ILOC
            IF(LTHL.GT.JERR) GO TO 10
                    IF(JKK.NE.4 .AND. JKK.NE.7) GO TO 15
                            JKL = JKK/IERR
                            GO TO 11
10          CONTINUE
                    JKL = JKK - 1
11          CONTINUE
                    J = MOD(JKL,IPLT)
   ITT=9
   CALL FPLEW(ITT)
   IF(IPRO.LE.0) GO TO 15
            J = MOD(JKL,IPRO)
   CALL FPRNT(ITT)
15              IF(IKEY.EQ.2) GO TO 20
   CALL INTRK5
                IF(BOOL) GO TO 40
```

```
      IF(JKK.EQ.LTHL) GO TO 30
      IF(IERRCT.EQ.IERR) GO TO 25
      IERRCT = IERRCT + 1
      GO TO 35
20 CALL ABAM
      XN = T2 - VIND
    IF(XN.LE.0.D0) GO TO 45
      IF(XN.GE.DELT) GO TO 35
      DELT = XN
      BOOL = .TRUE.
      IKEY = 1
      GO TO 35
25    IERRCT = 1
      I2B = I2B - 1
      GO TO 35
30    IKEY = 2
      DELT = STEP
      I48 = 4
      I2B = 3
       LTHL = (T2 - VIND)/DELT + 4.0D0*ERR
      GO TO 5
35    CONTINUE
40 CONTINUE
   K2B = I2B
45 CONTINUE
   RETURN
    END
```

```
      SUBROUTINE INTRK5
      PARAMETER N=8, NQ=4
      COMMON/EVAL/VIND,DELT,INDI
      DOUBLE PRECISION VIND,DELT
      COMMON/DIFEQ/P(N),DEPVAR(N),XN
      DOUBLE PRECISION P,DEPVAR,XN
      COMMON/DIRV/DEP(N,NQ),YPR(N,4,NQ)
      DOUBLE PRECISION DEP,YPR
      COMMON/MODE/S(N,3),JAY,NU,IONE,ITWO,IERRCT
      DOUBLE PRECISION S
      DOUBLE PRECISION XIP(N)
      COMMON/LOGIC/BOOL
      LOGICAL BOOL
      COMMON/COUNT/METHOD,ILOC,I1B,I2B,I3B,K2B
      DOUBLE PRECISION A(4), P1(4)
      DATA (A(J),J=1,4)/0.1666666666666667D0, 0.3333333333333333D0,
     1.3333333333333333D0,.1666666666666667D0/
      DATA (P1(J),J=1,4)/0.0D0,0.5D0,0.5D0,1.0D0/
      DATA IERR/3/
C
C     THIS ROUTINE PERFORMS A RUNGE-KUTTA INTEGRATION
C
      NS=NQ
            DO 90 JAY = 1,NS
               XN = VIND
               ISAV = INDI
               DO 10 J = 1,N
```

```
                XIP(J)=0.0D0
10                 DEPVAR(J) = DEP(J,JAY)
               NU = 1
15 CONTINUE
   IF(JAY.NE.1) GO TO 17
   CALL F1
   GO TO 18
17 CONTINUE
   CALL F2
18 CONTINUE
               IF(ISAV.EQ.2) GO TO 30
               ISAV = 2
               DO 20 J = 1,N
20                 YPR(J,4,JAY) = P(J)
30             DO 35 J = 1,N
35                 XIP(J) = A(NU)*P(J) + XIP(J)
               IF(NU - 4) 40, 55, 55
40             NU = NU + 1
               XN = VIND + P1(NU)*DELT
               DO 50 J = 1,N
50                 DEPVAR(J) = DEP(J,JAY) + P1(NU)*DELT*P(J)
               GO TO 15
55             DO 60 J = 1,N
                   DEP(J,JAY) = DEP(J,JAY) + DELT*XIP(J)
60                 DEPVAR(J) = DEP(J,JAY)
               IF(JAY.NE.1) GO TO 70
   CALL F1
               DO 65 J = 1,N
65                 S(J,ITWO) = DEPVAR(J)
```

```
                GO TO 75
70 CONTINUE
   CALL F2
                IF(BOOL) GO TO 76
75              IF(IERRCT.NE.IERR) GO TO 90
76              DO 80 J = 1,N
80                 YPR(J,I2B,JAY) = P(J)
90              CONTINUE
             INDI = 2
             VIND = XN
             ISAV = IONE
             IONE = ITWO
             ITWO = ISAV
    RETURN
    END
```

```
      SUBROUTINE ITER
      PARAMETER N=8 , NQ=4
      PARAMETER NQ1=NQ-1, NP=N-NQ+1
      COMMON/MATRIX/A(NQ,NQ),H(NQ),XF(NQ)
      DOUBLE PRECISION A,H,XF
      COMMON/INITAL/DEPO(N),TO,TF,EPS
      DOUBLE PRECISION DEPO,TO,TF,EPS
C
C     THIS ROUTINE ADDS THE CORRECTION ON THE INITIAL DEPENDENT VARIABLE
C
C     AND FINAL TIME.
C
      TF = TF + H(NQ)
              DO 30 I=1,NQ1
               J = NP + I
                   DEPO(J) = DEPO(J) + H(I)
   30         CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MINVDP(A,N,K,X,J2,I2)
C*     MATRIX INVERSION ROUTINE-FORMULATED BY E. G. CLAYTON
C**    --- CALLING SEQUENCE --
C***    CALL MINVDP(A,N,E,K)
C****  A--SQUARE ARRAY (DOUBLE PRECISION) CONTAINING ORIGINAL MATRIX
C****  N--ORDER OF ORIGINAL MATRIX
C****  E--TEST CRITERION FOR NEAR ZERO DIVISOR (DOUBLE PRECISION)
C***   K--LOCATION FOR SINGULARITY OR ILL-CONDITION INDICATOR
C**      K=0 =) MATRIX NONSINGULAR.
C*       K=1 =) MATRIX SINGULAR (OR ILL-CONDITIONED)
      DOUBLE PRECISION A,X,BIGA,DIV  ,E
      DIMENSION A(N,N),X(N),J2(N),I2(N)
      DATA E/.1D-17/
C*    INITIALIZATION
      M=N
      K=0
      I2(1)=0
      J2(1)=0
C*    BEGIN COMPUTATION OF THE INVERSE
      DO 15 L=1,M
      L1=L-1
      BIGA=0.0D0
C*    LOOK FOR THE ELEMENT OF GREATEST ABSOLUTE VALUE,CHOOSING
C*    ONE FROM A ROW AND COLUMN NOT PREVIOUSLY USED.
      DO 5 I=1,M
      DO 1 I3=1,L1
      IF(I-I2(I3))1,5,1
```

```
    1 CONTINUE
      DO 4 J=1,M
      DO 2 I3=1,L1
      IF(J-J2(I3))2,4,2
    2 CONTINUE
      IF(BIGA-DABS(A(I,J)))3,3,4
    3 BIGA=DABS(A(I,J))
      J1=J
      I1=I
    4 CONTINUE
    5 CONTINUE
C*    TAG THE ROW AND COLUMN FROM WHICH THE ELEMENT IS CHOSEN.
      J2(L)=J1
      I2(L)=I1
      DIV=A(I1,J1)
C*    TEST ELEMENT AGAINST ZERO CRITERION.
      IF(DABS(DIV)-E)221,221,6
C*    PERFORM THE COMPUTATIONS
    6 DO 7 J=1,M
      A(I1,J)=A(I1,J)/DIV
    7 CONTINUE
      A(I1,J1)=1.0D0/DIV
      DO 11 I=1,M
      IF(I1-I)8,11,8
    8 DO 10 J=1,M
      IF(J1-J)9,10,9
    9 A(I,J)=A(I,J)-A(I1,J)*A(I,J1)
   10 CONTINUE
   11 CONTINUE
```

```
      DO 14 I=1,M
      IF(I1-I) 13,14,13
   13 A(I,J1)=-A(I,J1)*A(I1,J1)
   14 CONTINUE
   15 CONTINUE
C*    COMPUTATION COMPLETE AT THIS POINT
C*    UNSCRAMBLE THE INVERSE
      DO 18 J=1,M
      DO 16 I=1,M
      I1=I2(I)
      J1=J2(I)
      X(J1)=A(I1,J)
   16 CONTINUE
      DO 17 I=1,M
      A(I,J)=X(I)
   17 CONTINUE
   18 CONTINUE
      DO 21 I=1,M
      DO 19 J=1,M
      I1=I2(J)
      J1=J2(J)
      X(I1)=A(I,J1)
   19 CONTINUE
      DO 20 J=1,M
      A(I,J)=X(J)
   20 CONTINUE
   21 CONTINUE
      RETURN
  221 K=1
```

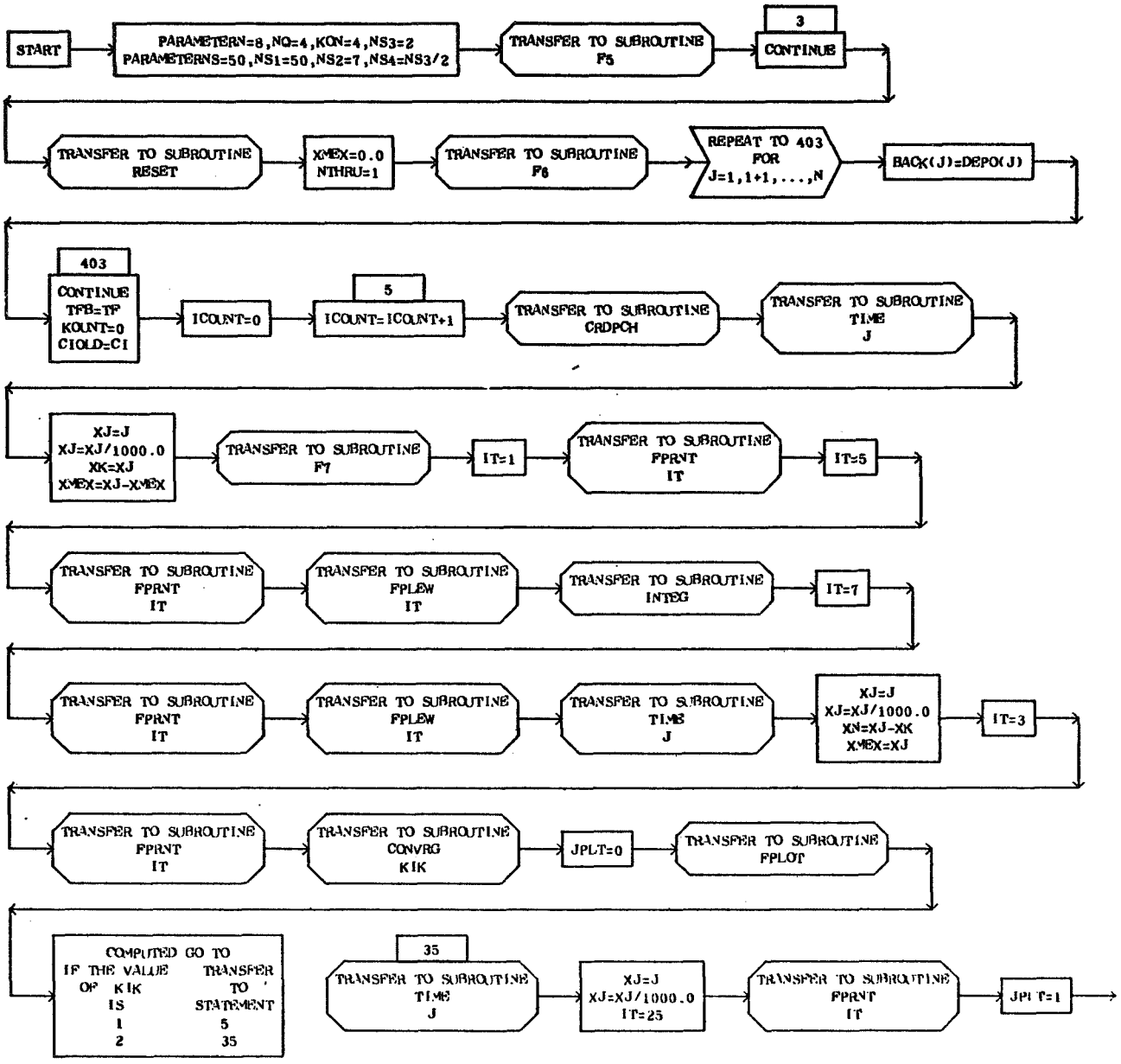
```
      RETURN
      END
```

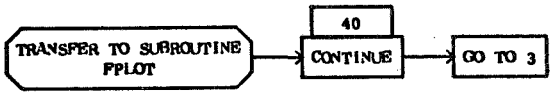
```
      SUBROUTINE MLTPLY
      PARAMETER NQ=4
      COMMON/FLAG/FL1,FL2,IFL
      DOUBLE PRECISION FL1,FL2
      COMMON/LEWALN/HNN,HMAX,NTHRU
      DOUBLE PRECISION HNN,HMAX
      COMMON/MATRIX/A(NQ,NQ),C(NQ),XF(NQ)
      DOUBLE PRECISION A,C,XF
      COMMON/TRL/D(NQ),ISET
      DOUBLE PRECISION D
      DOUBLE PRECISION B(NQ,NQ),BETA,BMAX,ALPHA
      DATA BETA/0.1D-03/
      DO 10 I=1,NQ
      DO 10 J=1,NQ
      B(I,J)=0.0D0
      DO 10 K=1,NQ
   10 B(I,J) = B(I,J) + A(K,I)*A(K,J)
      DO 35 I=1,NQ
      IF (B(I,I) .GT. BETA) GO TO 35
      BMAX=0.0D0
      DO 30 L=1,NQ
      IF (B(L,L) .GT. BMAX) BMAX = B(L,L)
   30 CONTINUE
      B(1,I) = BMAX
   35 CONTINUE
      GO TO (50,80,150,150),IFL
C STEPPED-ALPHA TECHNIQUE
```

```
   50 DO 60 I=1,NQ
   60 B(I,I) = B(I,I) + FL1*B(I,I)
      GO TO 300
C STEPPED-BETA TECHNIQUE
   80 DO 85 I=1,NQ
   85 B(I,I) = B(I,I) + FL1
      GO TO 300
C VARIABLE-ALPHA TECHNIQUE
  150 ALPHA = FL1 * (HNN/HMAX)**FL2
      IF (ALPHA .LT. 0.0D0) ALPHA = 0.0D0
      IF (ALPHA .GT. FL1) ALPHA = FL1
      IF (IFL .EQ. 4) GO TO 180
      DO 160 I=1,NQ
  160 B(I,I) = B(I,I) + ALPHA*B(I,I)
      GO TO 300
C VARIABLE-BETA TECHNIQUE
  180 DO 185 I=1,NQ
  185 B(I,I) = B(I,I) + ALPHA
C STORAGE MATRICES
  300 DO 310 I=1,NQ
      D(I) = 0.0
      DO 310 J=1,NQ
  310 D(I) = D(I) - A(J,I)*C(J)
      DO 330 I=1,NQ
      DO 330 J=1,NQ
  330 A(I,J) = B(I,J)
C
      RETURN
      END
```

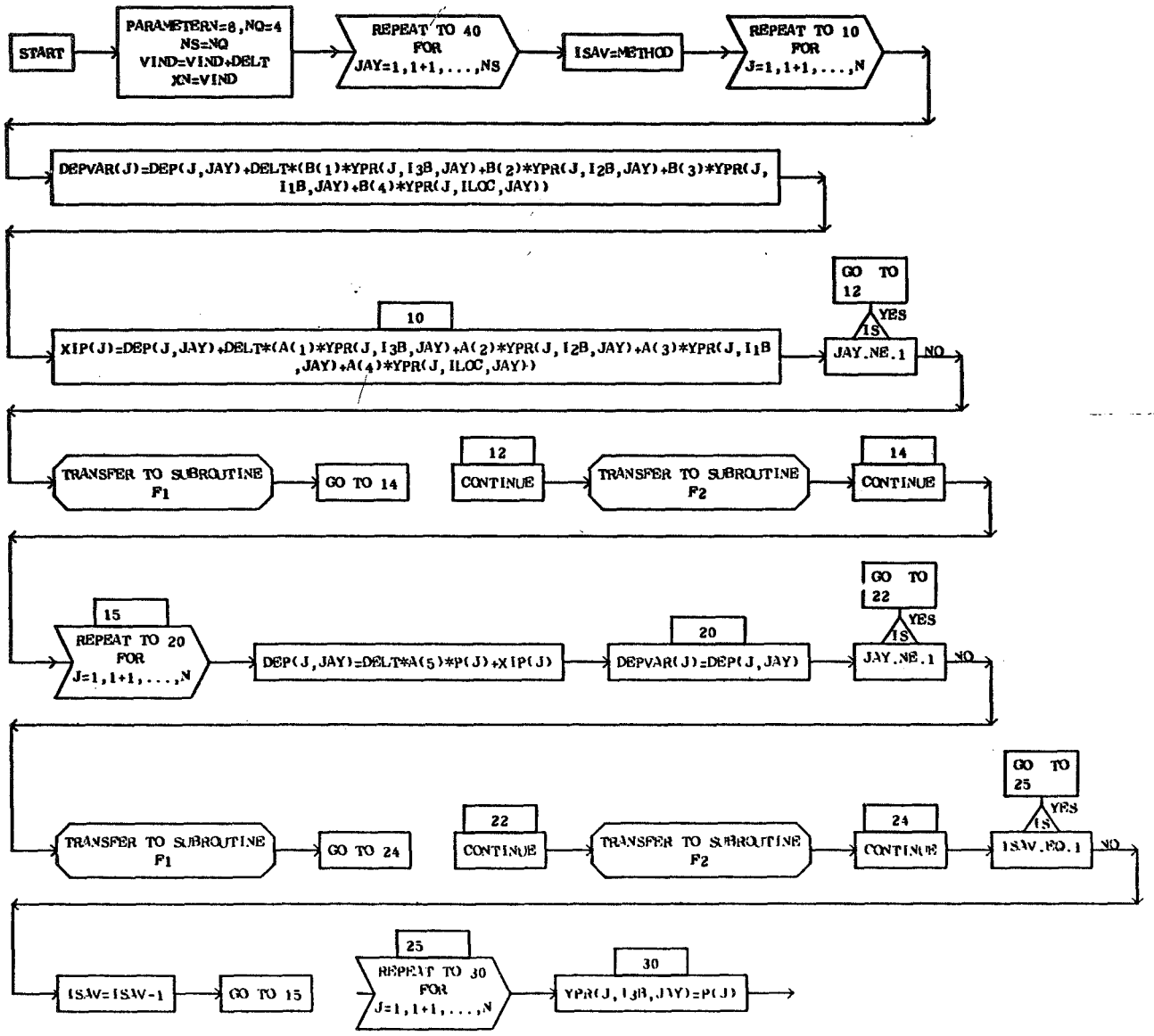
3. FLOW CHARTS

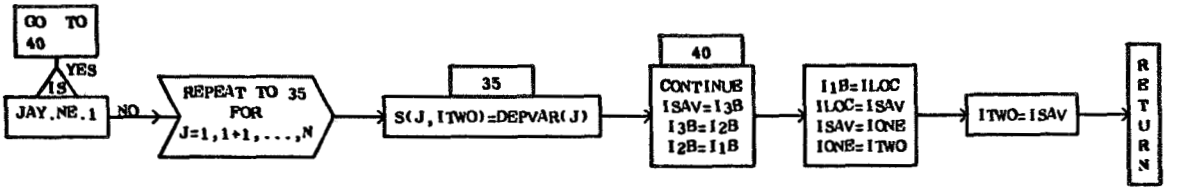
MAIN PROGRAM FOR MPF



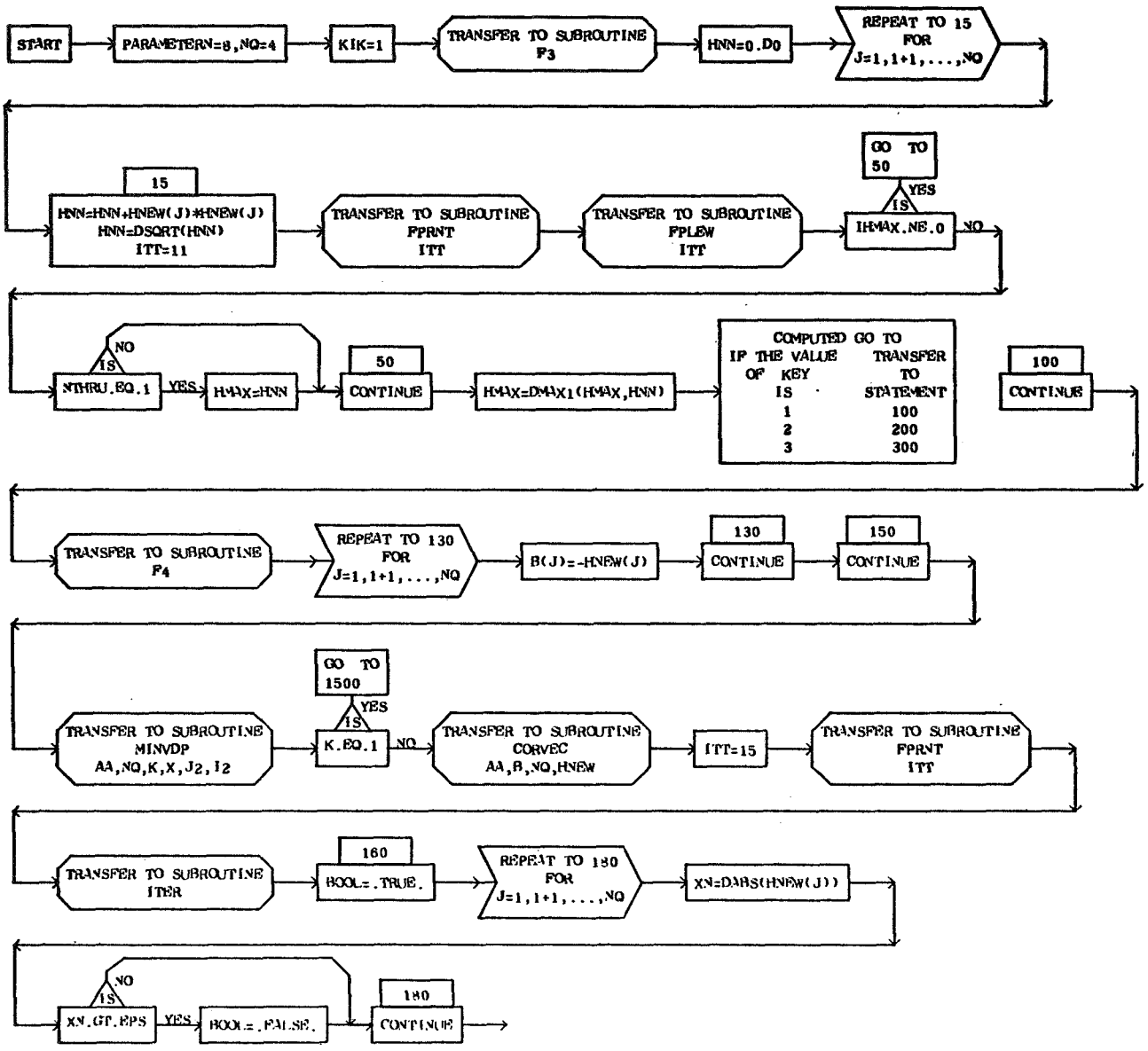

TRANSFER TO SUBROUTINE FPLOT
40
CONTINUE
GO TO 3

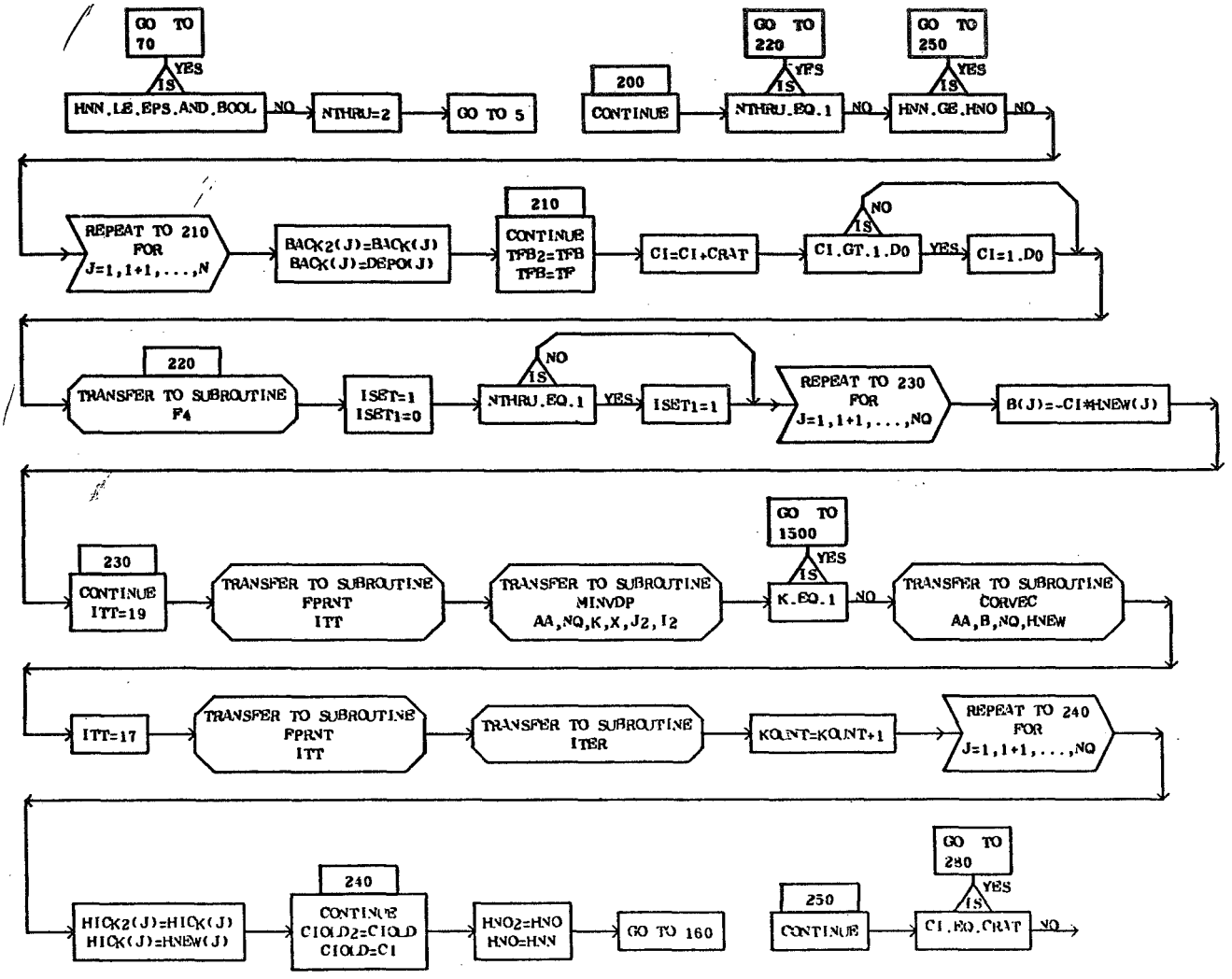
SUBROUTINE ABAM

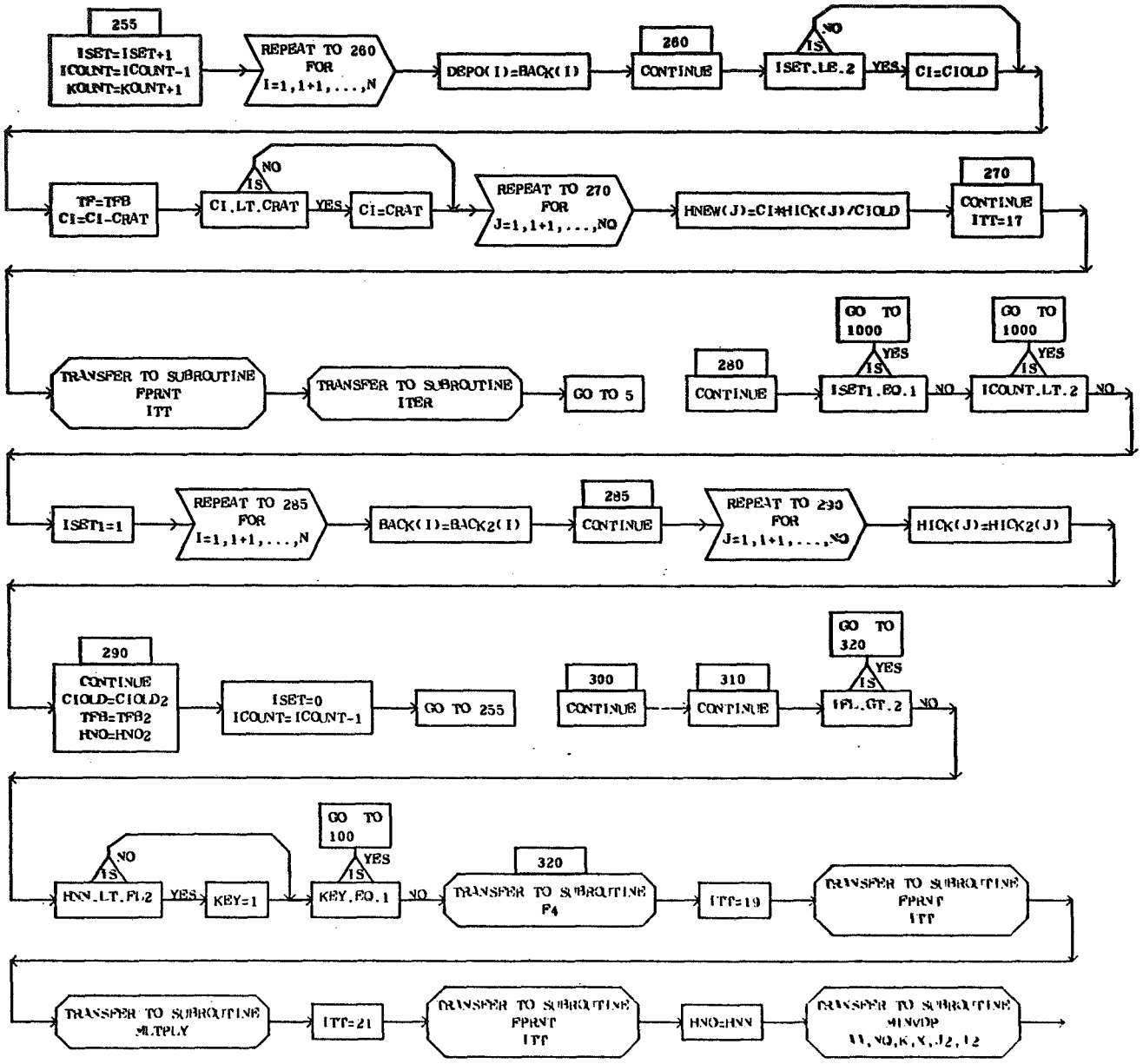


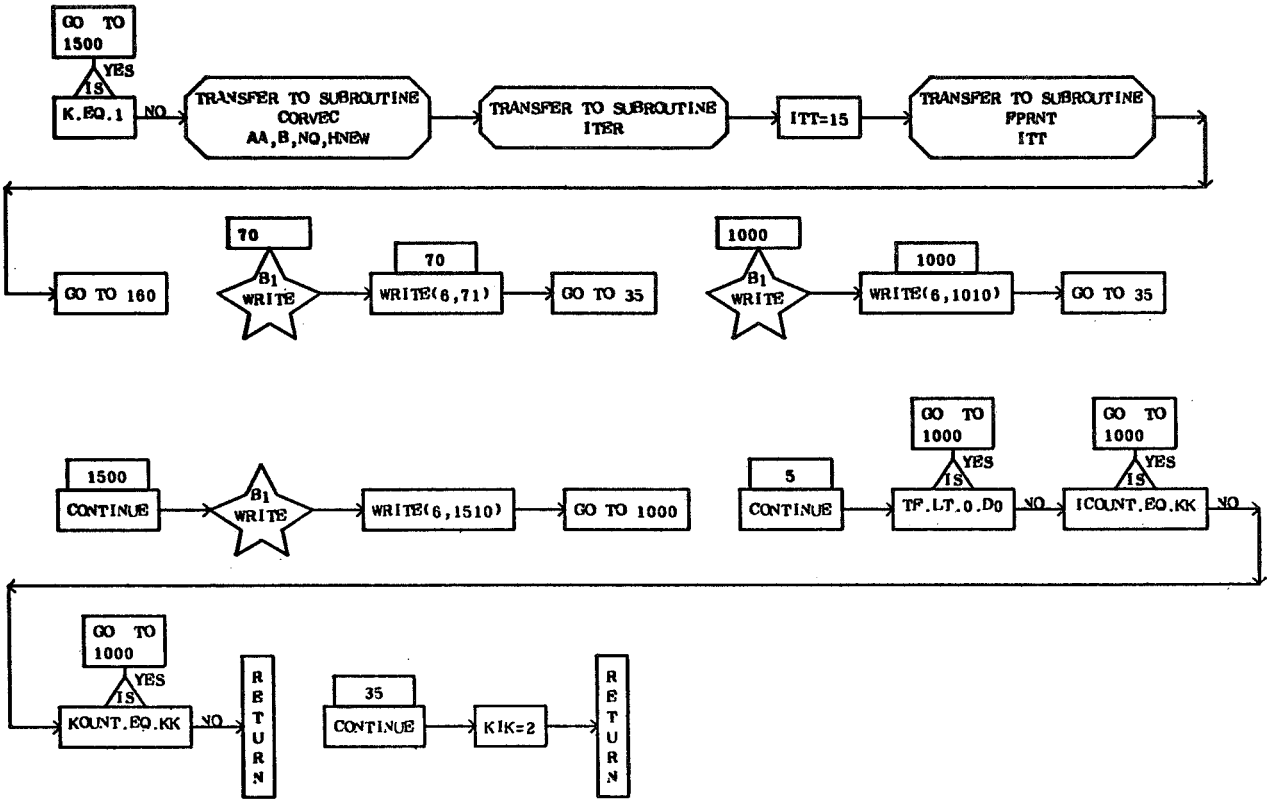

GO TO
40
YES
IS
JAY.NE.1
NO
REPEAT TO 35
FOR
J=1,1+1,...,N
35
S(J,ITWO)=DEPVAR(J)
40
CONTINUE
ISAV=I3B
I3B=I2B
I2B=I1B
I1B=ILOC
ILOC=ISAV
ISAV=IONE
IONE=ITWO
ITWO=ISAV
RETURN

SUBROUTINE CONVRG(KIK)

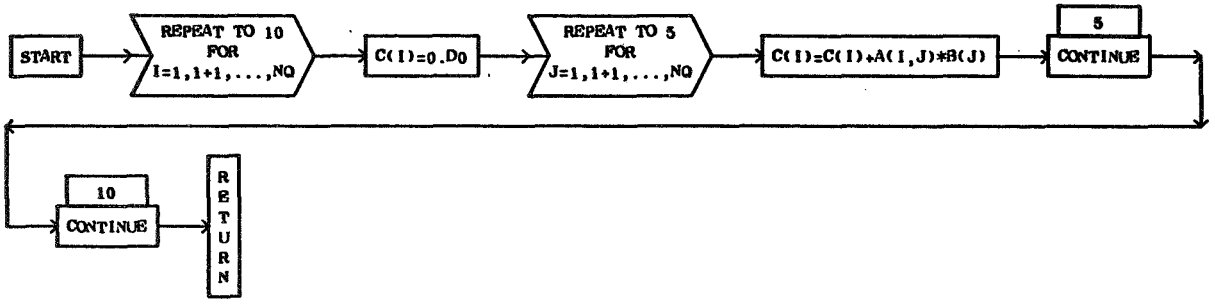


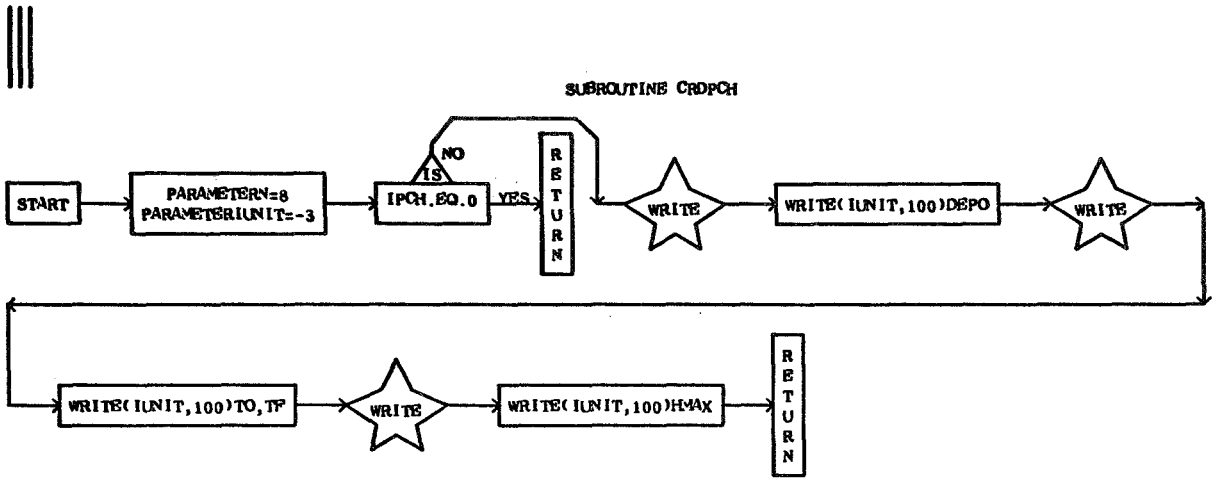

GO TO 70
IS
YES
HNN.LE.EPS.AND.BOOL
NO
NTHRU=2
GO TO 5
200
CONTINUE
GO TO 220
IS
YES
NTHRU.EQ.1
NO
GO TO 250
IS
YES
HNN.GE.HNO
NO
REPEAT TO 210 FOR J=1,1+1,...,N
BACK2(J)=BACK(J)
BACK(J)=DEPO(J)
210
CONTINUE
TFB2=TFB
TFB=TF
CI=CI+CRAT
IS
NO
CI.GT.1.D0
YES
CI=1.D0
220
TRANSFER TO SUBROUTINE F4
ISET=1
ISET1=0
IS
NO
NTHRU.EQ.1
YES
ISET1=1
REPEAT TO 230 FOR J=1,1+1,...,NQ
B(J)=-CI*HNEW(J)
230
CONTINUE
ITT=19
TRANSFER TO SUBROUTINE FPRNT ITT
TRANSFER TO SUBROUTINE MINVDP AA,NQ,K,X,J2,I2
GO TO 1500
IS
YES
K.EQ.1
NO
TRANSFER TO SUBROUTINE CORVEC AA,B,NQ,HNEW
ITT=17
TRANSFER TO SUBROUTINE FPRNT ITT
TRANSFER TO SUBROUTINE ITER
KOUNT=KOUNT+1
REPEAT TO 240 FOR J=1,1+1,...,NQ
HICK2(J)=HICK(J)
HICK(J)=HNEW(J)
240
CONTINUE
CIOLD2=CIOLD
CIOLD=CI
HNO2=HNO
HNO=HNN
GO TO 160
250
CONTINUE
GO TO 280
IS
YES
CI.EQ.CRAT
NO


255
ISET=ISET+1
ICOUNT=ICOUNT-1
KOUNT=KOUNT+1
REPEAT TO 260 FOR I=1,1+1,...,N
DEPO(I)=BACK(I)
260
CONTINUE
IS
NO
ISET.LE.2
YES
CI=CIOLD
TF=TFB
CI=CI-CRAT
IS
NO
CI.LT.CRAT
YES
CI=CRAT
REPEAT TO 270 FOR J=1,1+1,...,NO
HNEW(J)=CI*HICK(J)/CIOLD
270
CONTINUE
ITT=17
TRANSFER TO SUBROUTINE FPRNT ITT
TRANSFER TO SUBROUTINE ITER
GO TO 5
280
CONTINUE
GO TO 1000
YES
IS
ISET1.EQ.1
NO
GO TO 1000
YES
IS
ICOUNT.LT.2
NO
ISET1=1
REPEAT TO 285 FOR I=1,1+1,...,N
BACK(I)=BACK2(I)
285
CONTINUE
REPEAT TO 290 FOR J=1,1+1,...,NO
HICK(J)=HICK2(J)
290
CONTINUE
CIOLD=CIOLD2
TFB=TFB2
HNO=HNO2
ISET=0
ICOUNT=ICOUNT-1
GO TO 255
300
CONTINUE
310
CONTINUE
GO TO 320
YES
IS
IFL.GT.2
NO
IS
NO
HNN.LT.FL2
YES
KEY=1
GO TO 100
YES
IS
KEY.EQ.1
NO
320
TRANSFER TO SUBROUTINE P4
ITT=19
TRANSFER TO SUBROUTINE FPRNT ITT
TRANSFER TO SUBROUTINE MLTPLY
ITT=21
TRANSFER TO SUBROUTINE FPRNT ITT
HNO=HNN
TRANSFER TO SUBROUTINE MINVDP

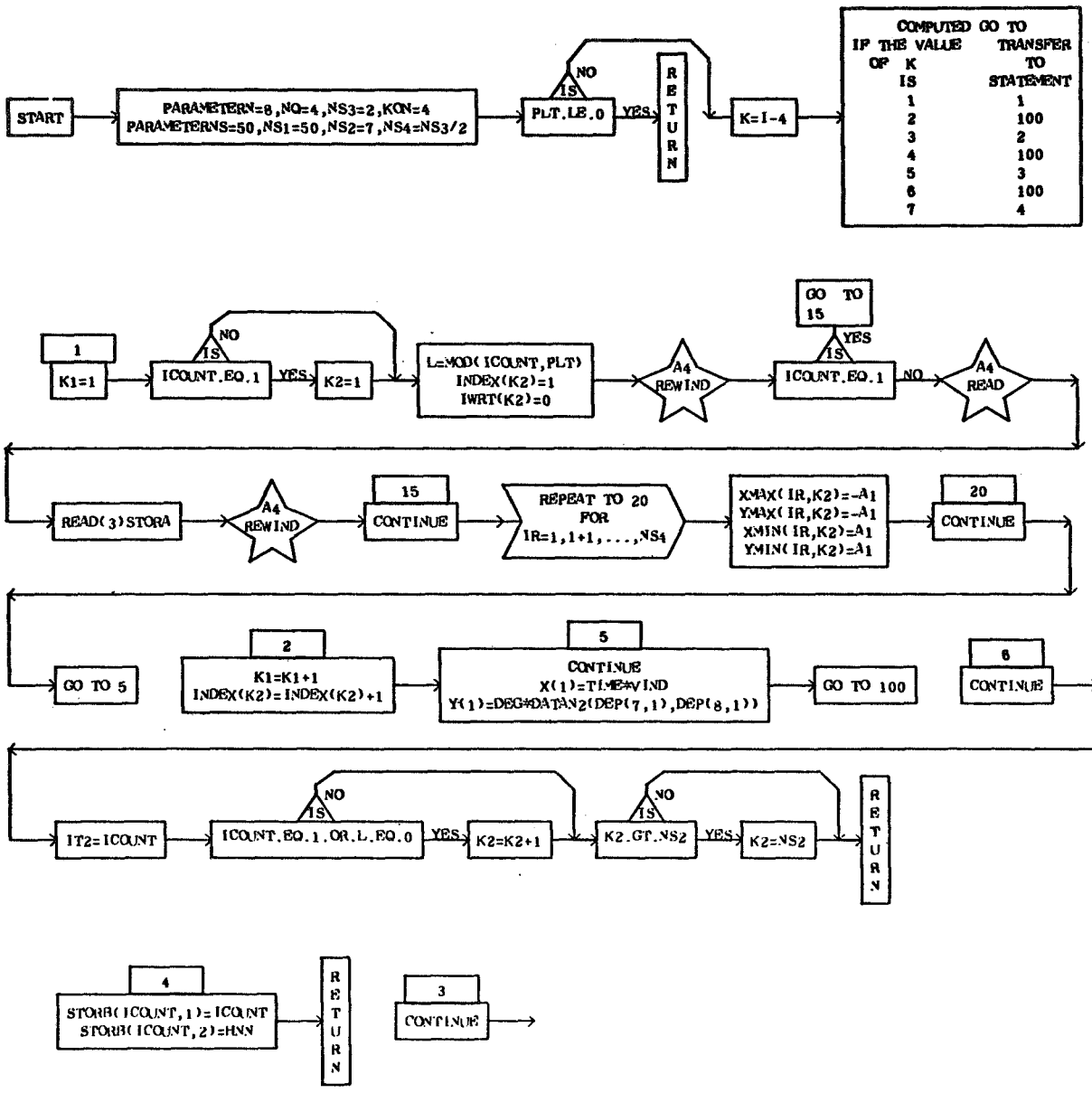

GO TO 1500
YES
IS
K.EQ.1
NO
TRANSFER TO SUBROUTINE CORVEC AA,B,NQ,HNEW
TRANSFER TO SUBROUTINE ITER
ITT=15
TRANSFER TO SUBROUTINE FPRNT ITT
GO TO 160
70
B1 WRITE
70
WRITE(6,71)
GO TO 35
1000
B1 WRITE
1000
WRITE(6,1010)
GO TO 35
1500
CONTINUE
B1 WRITE
WRITE(6,1510)
GO TO 1000
5
CONTINUE
GO TO 1000
YES
IS
TF.LT.0.D0
NO
GO TO 1000
YES
IS
ICOUNT.EQ.KK
NO
GO TO 1000
YES
IS
KOUNT.EQ.KK
NO
RETURN
35
CONTINUE
KIK=2
RETURN

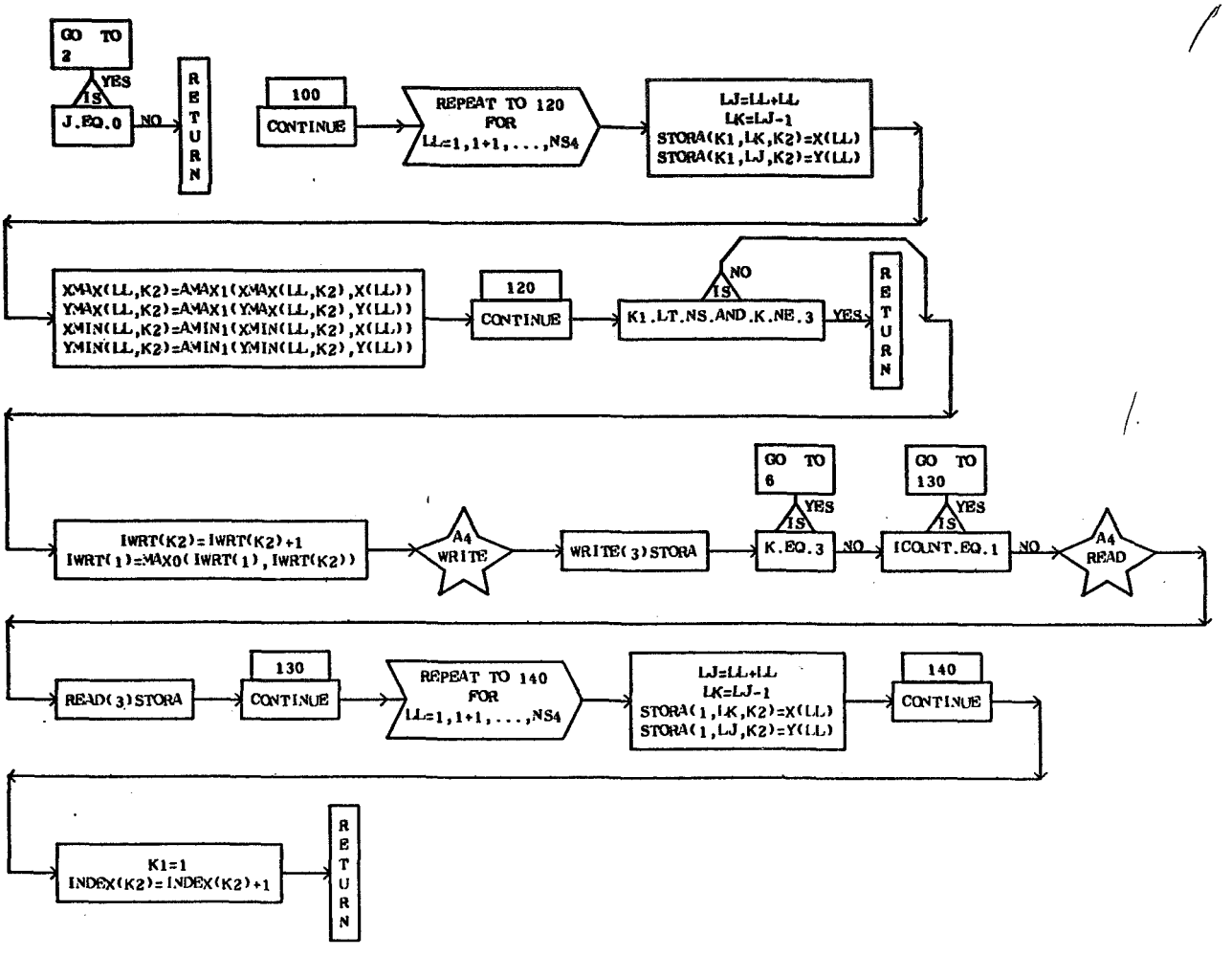
SUBROUTINE CORVEC(A,B,NQ,C)



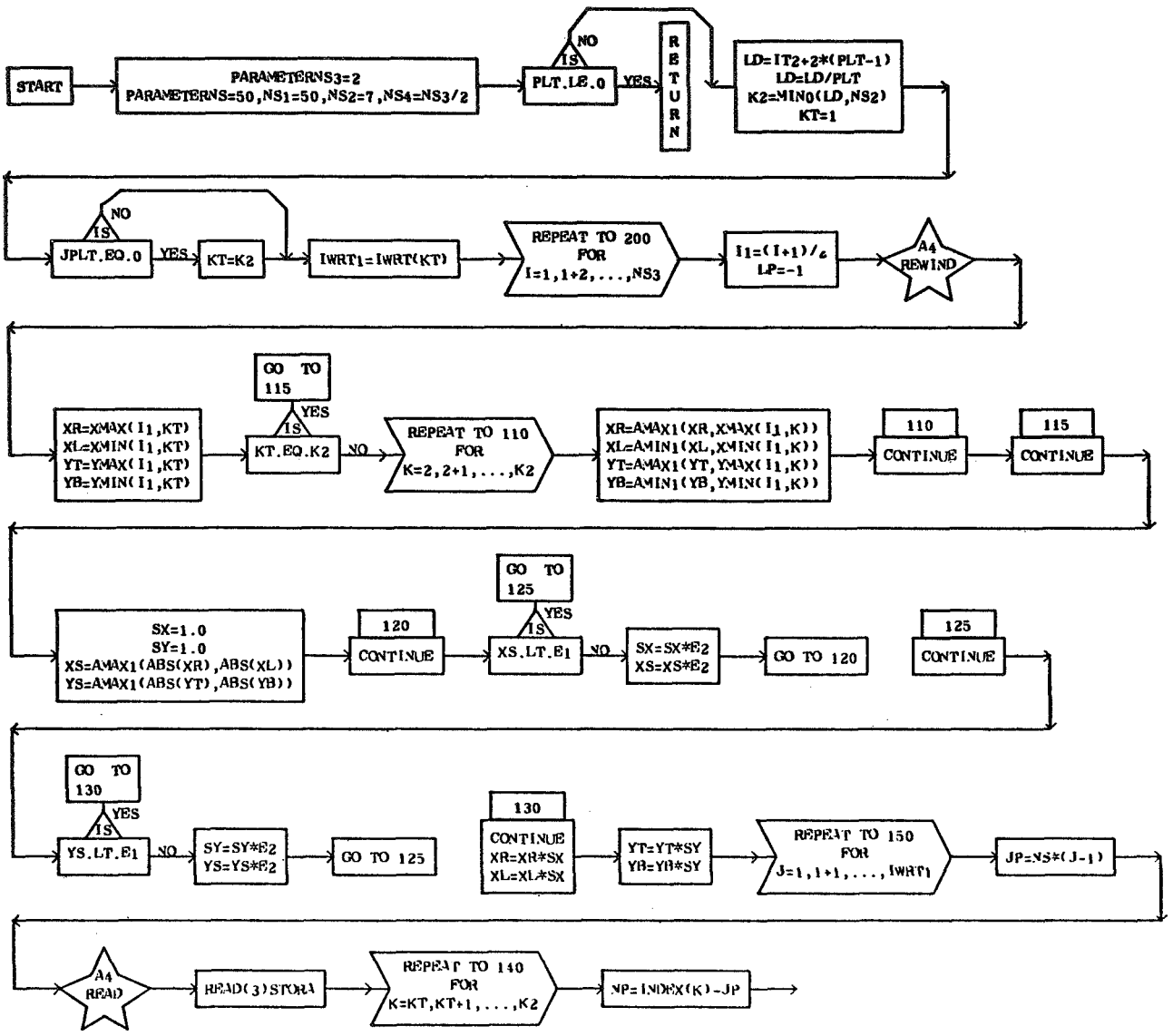

SUBROUTINE CRDPCH
START
PARAMETERN=8
PARAMETERIUNIT=-3
IPCH.EQ.0
IS
NO
YES
RETURN
WRITE
WRITE(IUNIT,100)DEPO
WRITE
WRITE(IUNIT,100)TO,TF
WRITE
WRITE(IUNIT,100)HMAX
RETURN

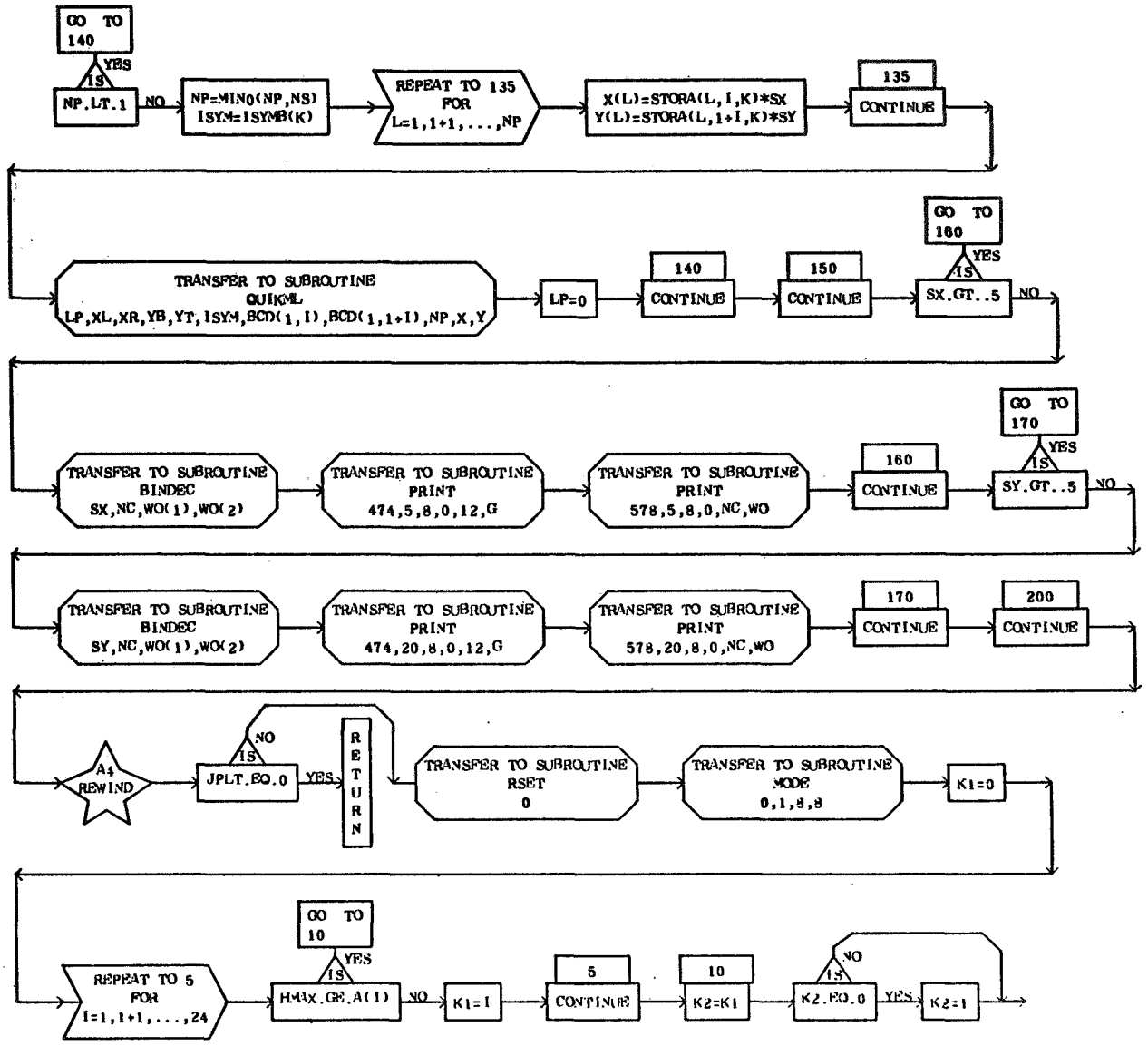
SUBROUTINE FPLEW(I)

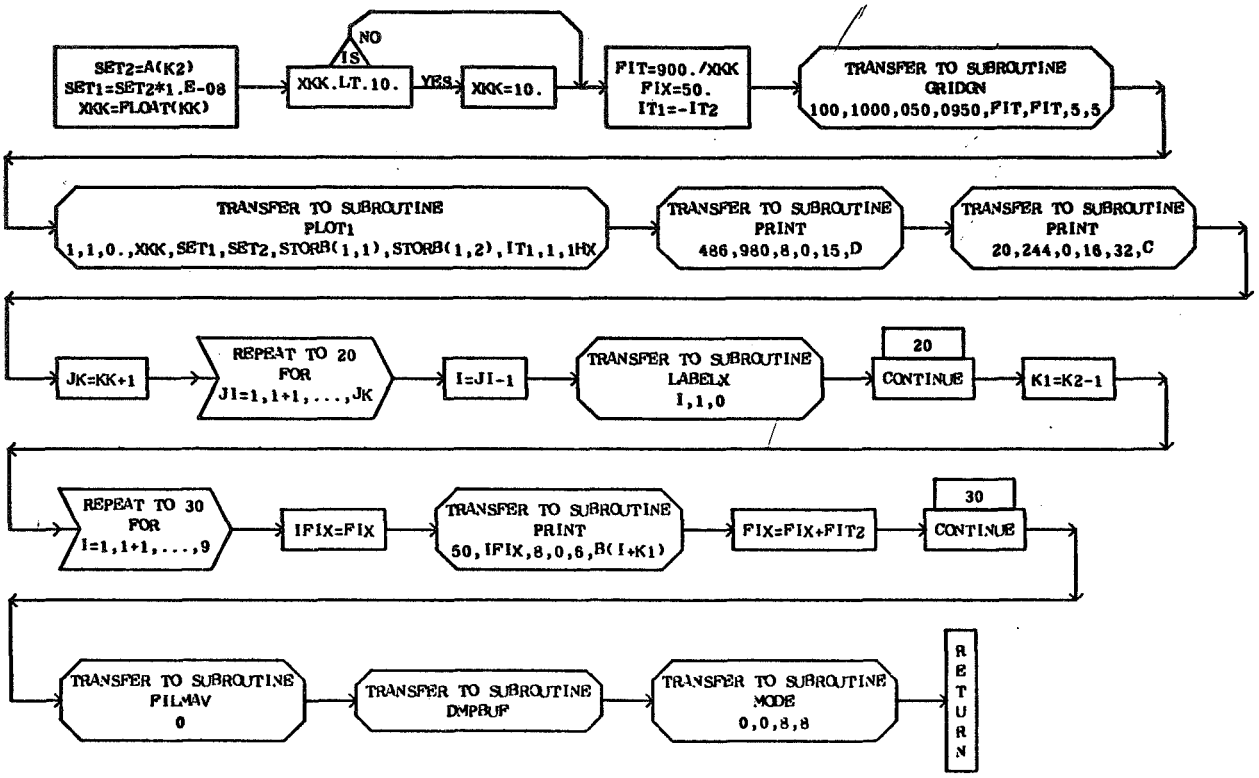



GO TO 2
IS
YES
J.EQ.0
NO
RETURN
100
CONTINUE
REPEAT TO 120 FOR LL=1,1+1,...,NS4
LJ=LL+LL
LK=LJ-1
STORA(K1,LK,K2)=X(LL)
STORA(K1,LJ,K2)=Y(LL)
XMAX(LL,K2)=AMAX1(XMAX(LL,K2),X(LL))
YMAX(LL,K2)=AMAX1(YMAX(LL,K2),Y(LL))
XMIN(LL,K2)=AMIN1(XMIN(LL,K2),X(LL))
YMIN(LL,K2)=AMIN1(YMIN(LL,K2),Y(LL))
120
CONTINUE
NO
IS
K1.LT.NS.AND.K.NE.3
YES
RETURN
IWRT(K2)=IWRT(K2)+1
IWRT(1)=MAX0(IWRT(1),IWRT(K2))
A4 WRITE
WRITE(3)STORA
GO TO 6
YES
IS
K.EQ.3
NO
GO TO 130
YES
IS
ICOUNT.EQ.1
NO
A4 READ
READ(3)STORA
130
CONTINUE
REPEAT TO 140 FOR LL=1,1+1,...,NS4
LJ=LL+LL
LK=LJ-1
STORA(1,LK,K2)=X(LL)
STORA(1,LJ,K2)=Y(LL)
140
CONTINUE
K1=1
INDEX(K2)=INDEX(K2)+1
RETURN

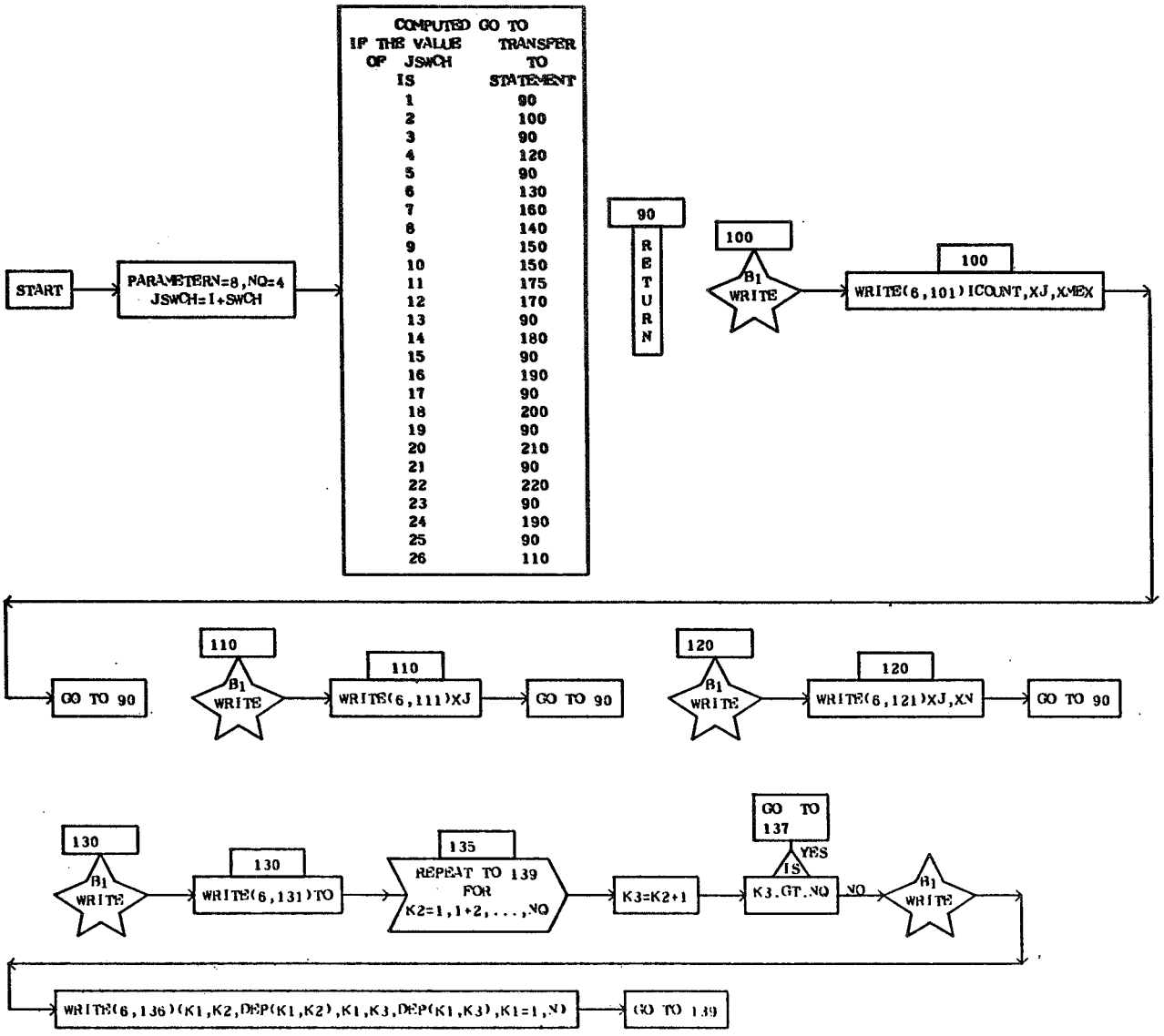
SUBROUTINE FPLOT

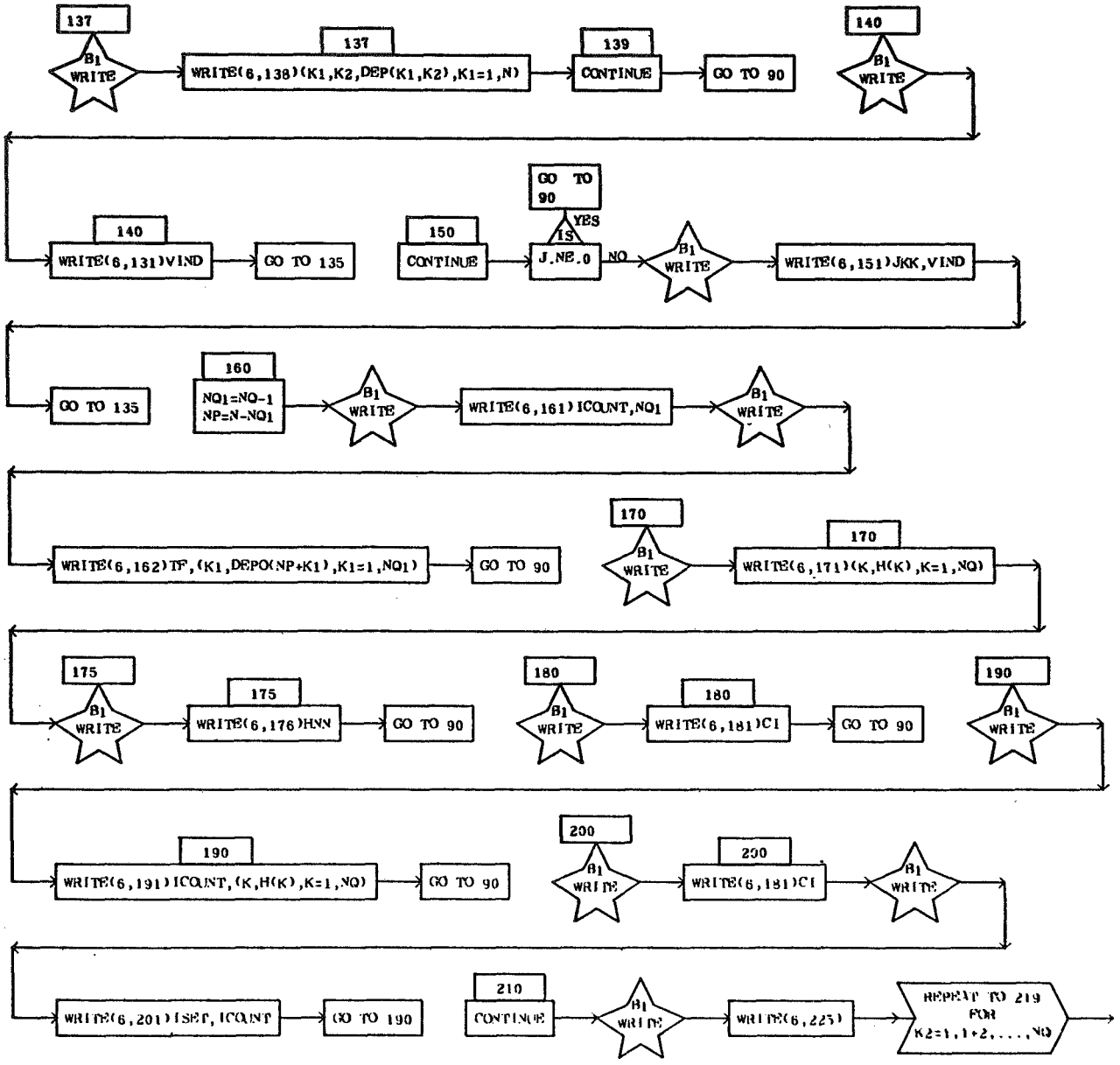


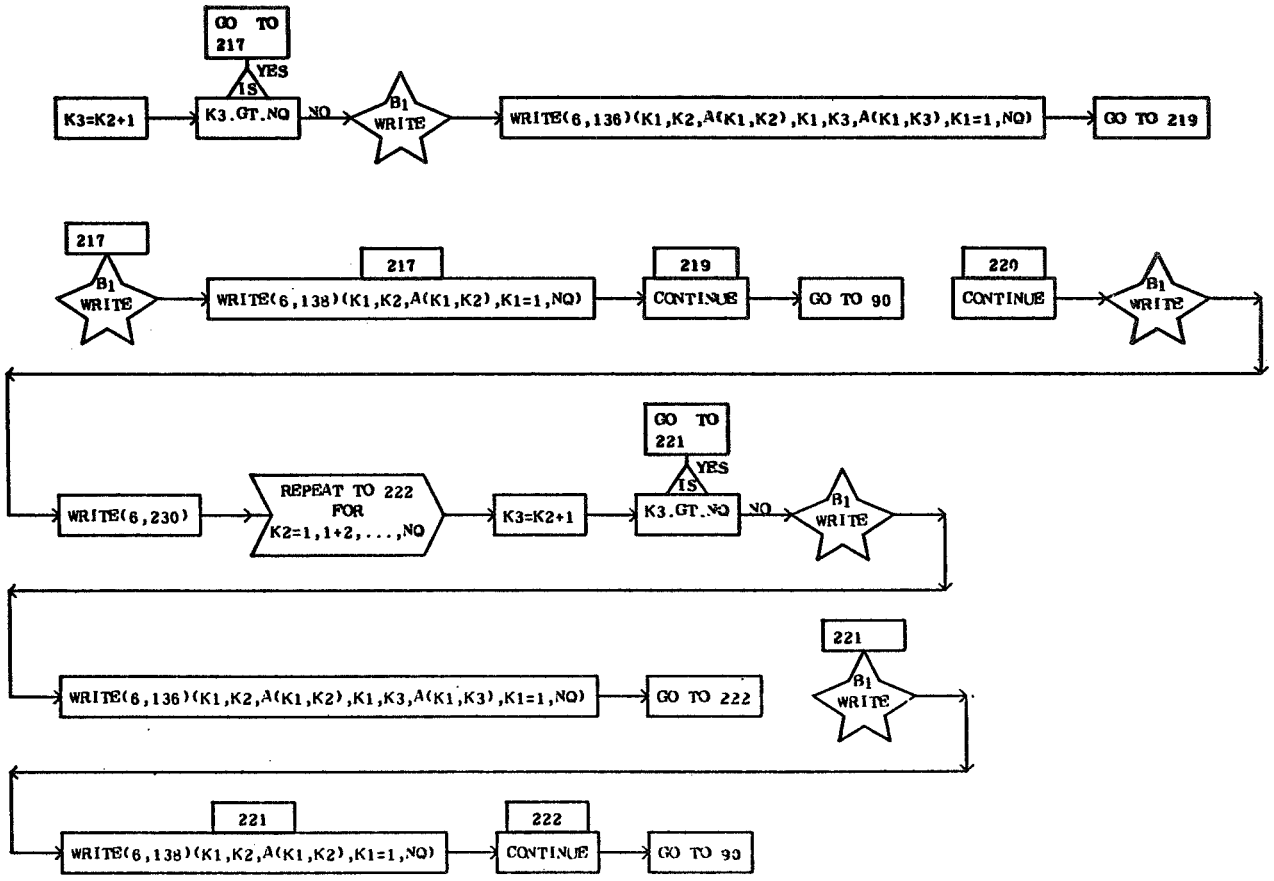

GO TO 140
IS
YES
NP.LT.1
NO
NP=MIN0(NP,NS)
ISYM=ISYMB(K)
REPEAT TO 135
FOR
L=1,1+1,...,NP
X(L)=STORA(L,I,K)*SX
Y(L)=STORA(L,1+I,K)*SY
135
CONTINUE
TRANSFER TO SUBROUTINE
QUIKML
LP,XL,XR,YB,YT,ISYM,BCD(1,I),BCD(1,1+I),NP,X,Y
LP=0
140
CONTINUE
150
CONTINUE
GO TO 160
IS
YES
SX.GT..5
NO
TRANSFER TO SUBROUTINE
BINDEC
SX,NC,WO(1),WO(2)
TRANSFER TO SUBROUTINE
PRINT
474,5,8,0,12,G
TRANSFER TO SUBROUTINE
PRINT
578,5,8,0,NC,WO
160
CONTINUE
GO TO 170
IS
YES
SY.GT..5
NO
TRANSFER TO SUBROUTINE
BINDEC
SY,NC,WO(1),WO(2)
TRANSFER TO SUBROUTINE
PRINT
474,20,8,0,12,G
TRANSFER TO SUBROUTINE
PRINT
578,20,8,0,NC,WO
170
CONTINUE
200
CONTINUE
A4
REWIND
IS
NO
JPLT.EQ.0
YES
RETURN
TRANSFER TO SUBROUTINE
RSET
0
TRANSFER TO SUBROUTINE
MODE
0,1,8,8
K1=0
REPEAT TO 5
FOR
I=1,1+1,...,24
GO TO 10
IS
YES
HMAX.GE.A(I)
NO
K1=I
5
CONTINUE
10
K2=K1
IS
NO
K2.EQ.0
YES
K2=1


SET2=A(K2)
SET1=SET2*1.E-08
XKK=FLOAT(KK)
XKK.LT.10.
IS
NO
YES
XKK=10.
FIT=900./XKK
FIX=50.
IT1=-IT2
TRANSFER TO SUBROUTINE
GRIDGN
100,1000,050,0950,FIT,FIT,5,5
TRANSFER TO SUBROUTINE
PLOT1
1,1,0.,XKK,SET1,SET2,STORB(1,1),STORB(1,2),IT1,1,1HX
TRANSFER TO SUBROUTINE
PRINT
486,980,8,0,15,D
TRANSFER TO SUBROUTINE
PRINT
20,244,0,16,32,C
JK=KK+1
REPEAT TO 20
FOR
JI=1,1+1,...,JK
I=JI-1
TRANSFER TO SUBROUTINE
LABELX
I,1,0
20
CONTINUE
K1=K2-1
REPEAT TO 30
FOR
I=1,1+1,...,9
IFIX=FIX
TRANSFER TO SUBROUTINE
PRINT
50,IFIX,8,0,6,B(I+K1)
FIX=FIX+FIT2
30
CONTINUE
TRANSFER TO SUBROUTINE
FILMAV
0
TRANSFER TO SUBROUTINE
DMPBUF
TRANSFER TO SUBROUTINE
MODE
0,0,8,8
RETURN

SUBROUTINE FPRNT(I)




137
B1 WRITE
137
WRITE(6,138)(K1,K2,DEP(K1,K2),K1=1,N)
139
CONTINUE
GO TO 90
140
B1 WRITE
140
WRITE(6,131)VIND
GO TO 135
150
CONTINUE
GO TO 90
IS
YES
J.NE.0
NO
B1 WRITE
WRITE(6,151)JKK,VIND
GO TO 135
160
NQ1=NQ-1
NP=N-NQ1
B1 WRITE
WRITE(6,161)ICOUNT,NQ1
B1 WRITE
WRITE(6,162)TF,(K1,DEPO(NP+K1),K1=1,NQ1)
GO TO 90
170
B1 WRITE
170
WRITE(6,171)(K,H(K),K=1,NQ)
175
B1 WRITE
175
WRITE(6,176)HNN
GO TO 90
180
B1 WRITE
180
WRITE(6,181)CI
GO TO 90
190
B1 WRITE
190
WRITE(6,191)ICOUNT,(K,H(K),K=1,NQ)
GO TO 90
200
B1 WRITE
200
WRITE(6,181)CI
B1 WRITE
WRITE(6,201)ISET,ICOUNT
GO TO 190
210
CONTINUE
B1 WRITE
WRITE(6,225)
REPEAT TO 219
FOR
K2=1,1+2,...,NQ


GO TO 217
YES
IS
K3=K2+1
K3.GT.NQ
NO
B1 WRITE
WRITE(6,136)(K1,K2,A(K1,K2),K1,K3,A(K1,K3),K1=1,NQ)
GO TO 219
217
B1 WRITE
217
WRITE(6,138)(K1,K2,A(K1,K2),K1=1,NQ)
219
CONTINUE
GO TO 90
220
CONTINUE
B1 WRITE
WRITE(6,230)
REPEAT TO 222 FOR K2=1,1+2,...,NQ
K3=K2+1
GO TO 221
YES
IS
K3.GT.NQ
NO
B1 WRITE
WRITE(6,136)(K1,K2,A(K1,K2),K1,K3,A(K1,K3),K1=1,NQ)
GO TO 222
221
B1 WRITE
221
WRITE(6,138)(K1,K2,A(K1,K2),K1=1,NQ)
222
CONTINUE
GO TO 90

SUBROUTINE F1

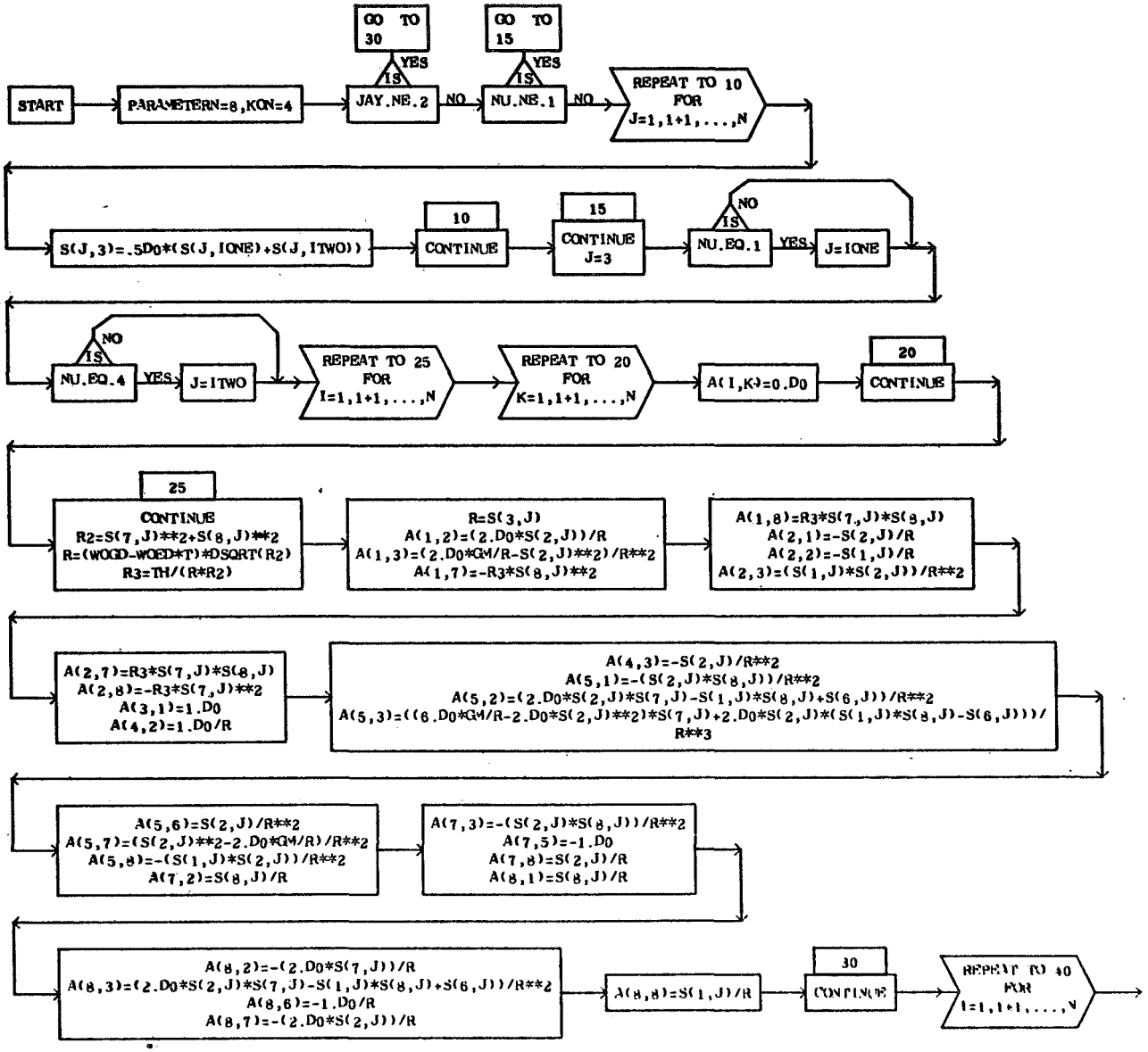
```
START
PARAMETERN=8,KON=4
XNN=(WOGD-WOED*T)*DSQRT(TY(7)**2+TY(8)**2)
P(1)=TY(2)**2/TY(3)-GM/TY(3)**2-TH*TY(7)/XNN
P(2)=-TY(1)*TY(2)/TY(3)-TH*TY(8)/XNN
```

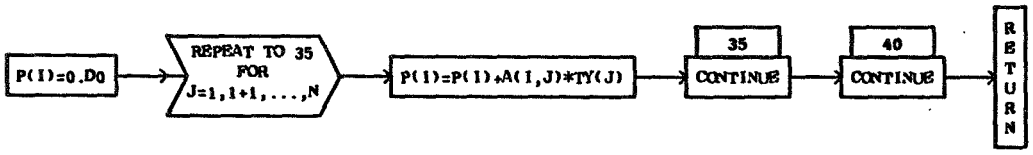
```
P(3)=TY(1)
P(4)=TY(2)/TY(3)
P(5)=(((TY(2)**2-2.D0*GM/TY(3))*TY(7))-(TY(1)*TY(2)*TY(8))+(TY(2)*TY(6)))/(TY(
3)**2)
```

```
P(6)=0.D0
P(7)=TY(2)*TY(8)/TY(3)-TY(5)
P(8)=(-2.D0*TY(2)*TY(7)+TY(1)*TY(8)-TY(6))/TY(3)
```

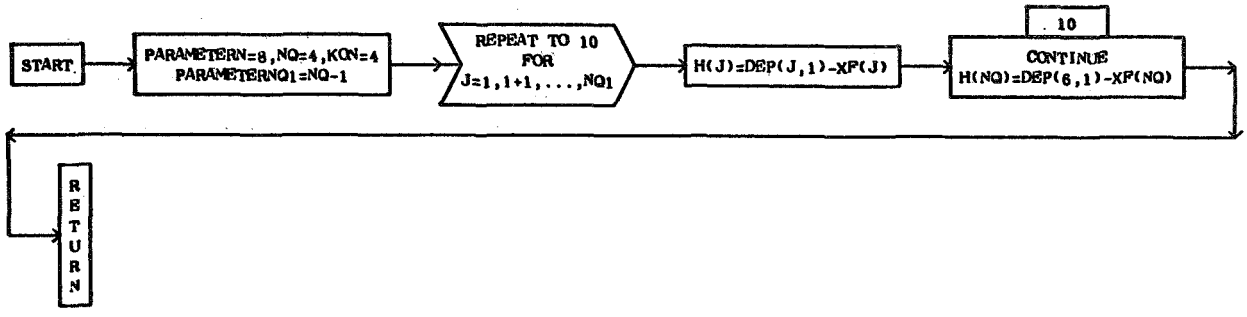
RETURN

SUBROUTINE F2

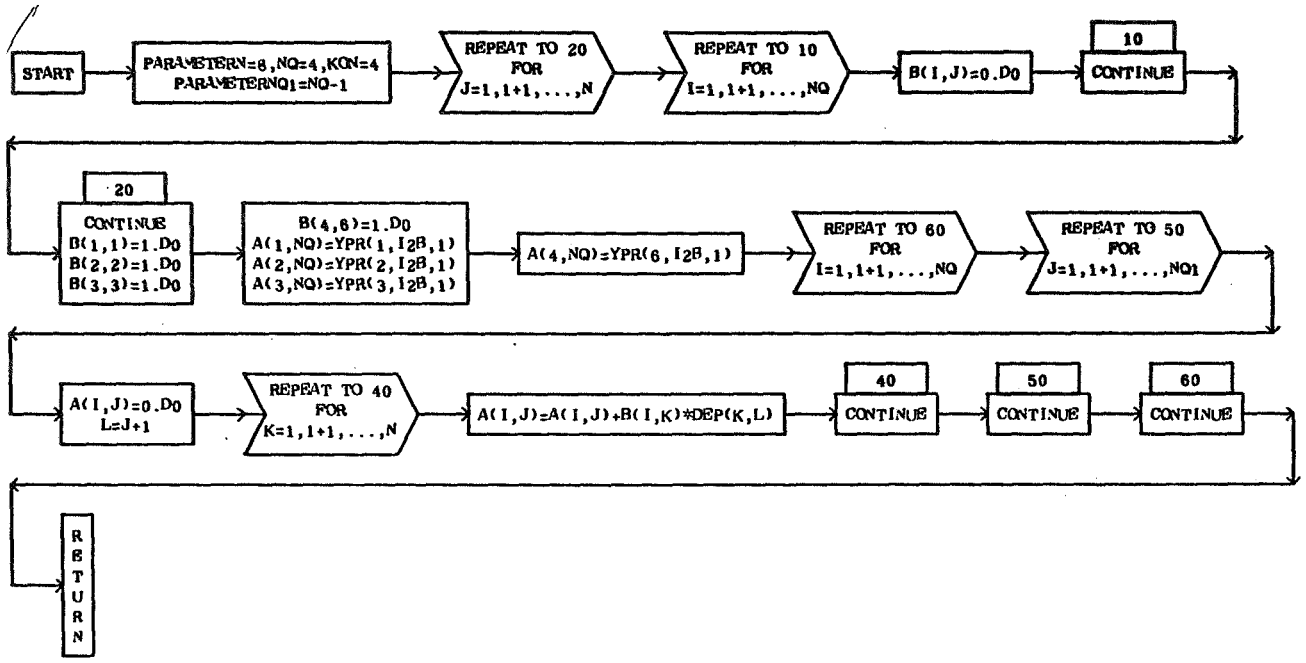



P(I)=0.D0
REPEAT TO 35
FOR
J=1,I+1,...,N
P(I)=P(I)+A(I,J)*TY(J)
35
CONTINUE
40
CONTINUE
RETURN

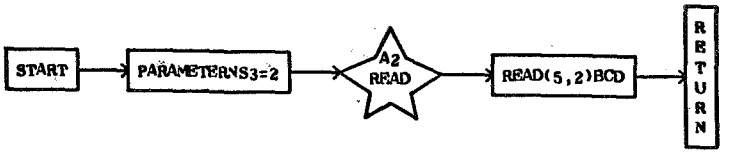
SUBROUTINE F3



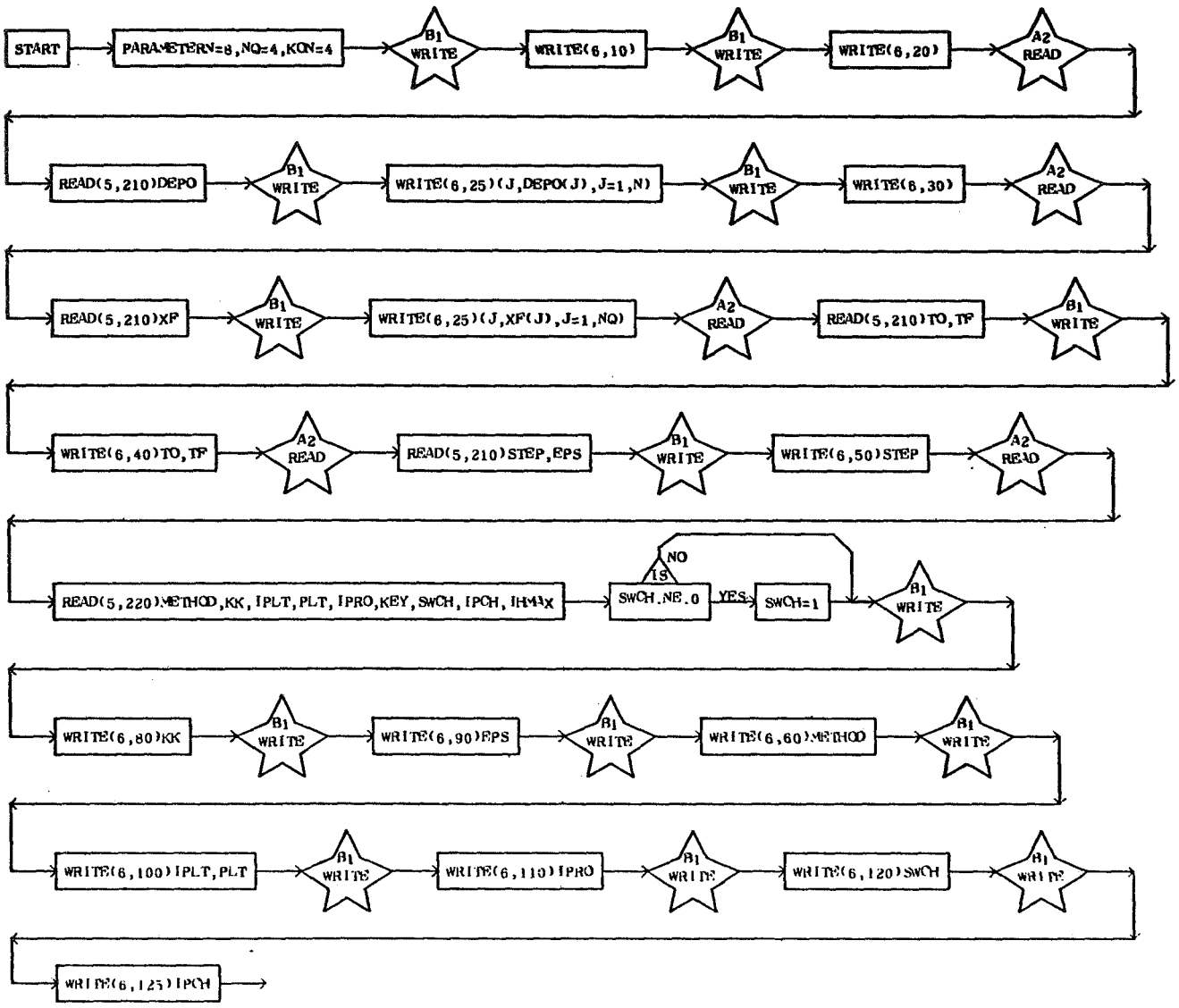
SUBROUTINE F4

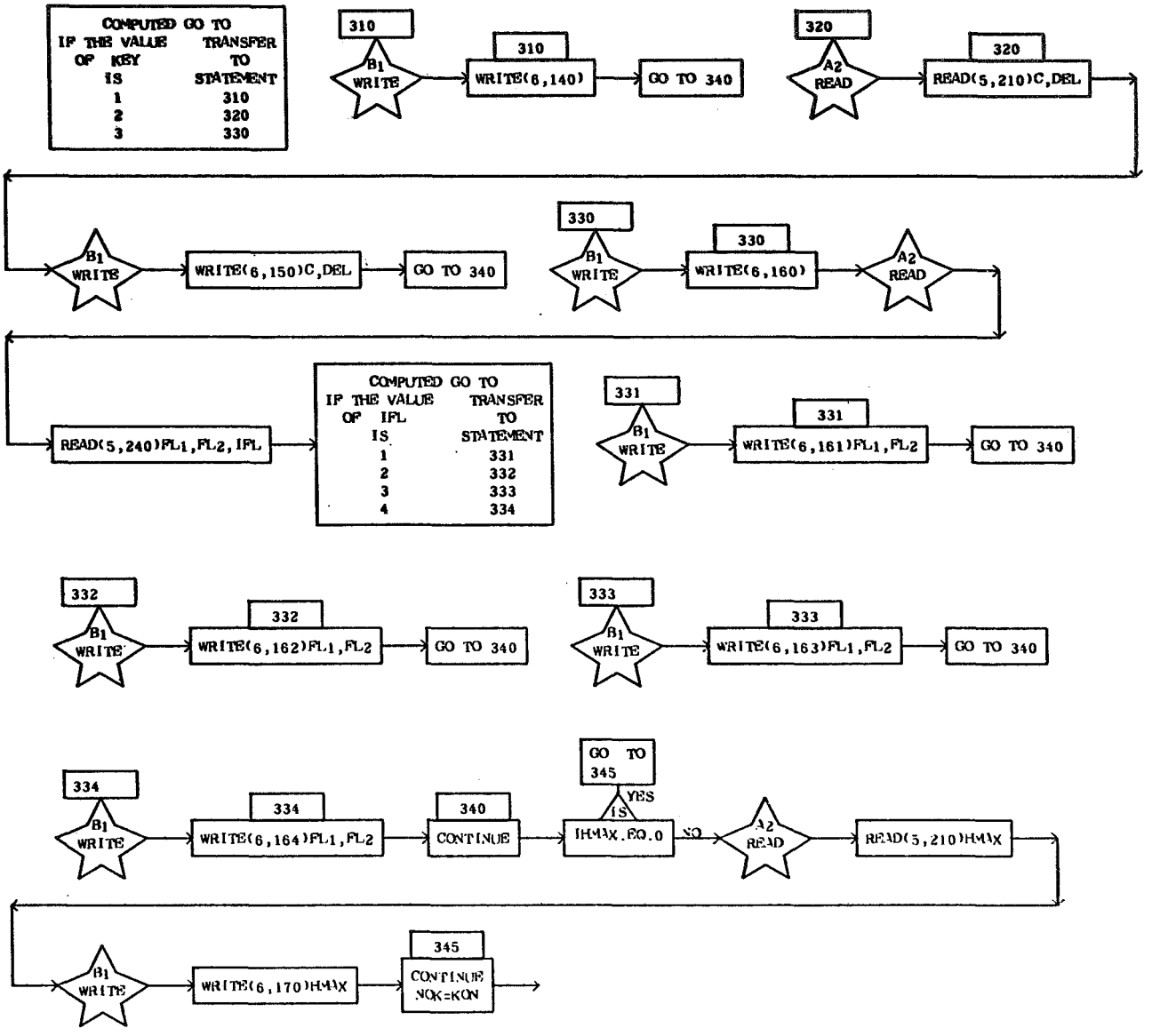


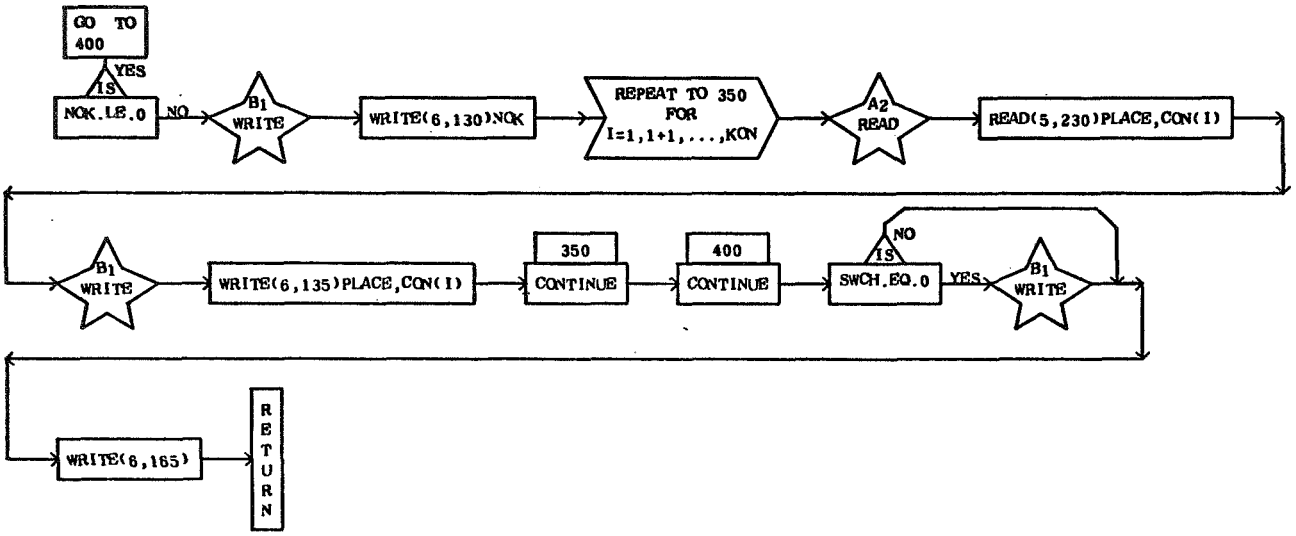
SUBROUTINE F5



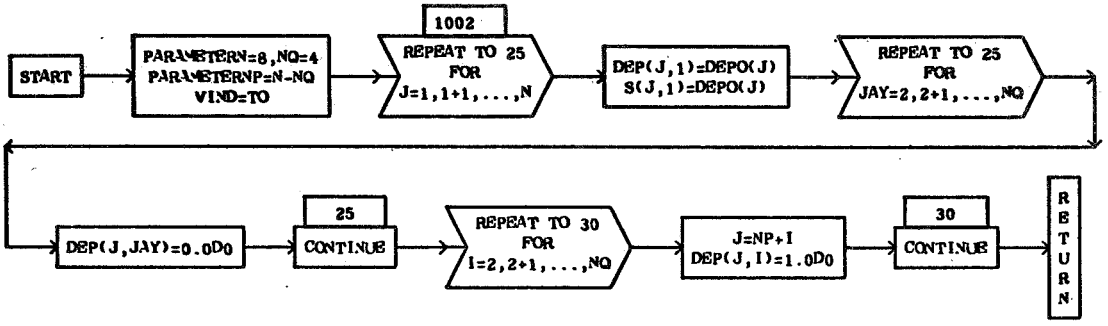
SUBROUTINE F6



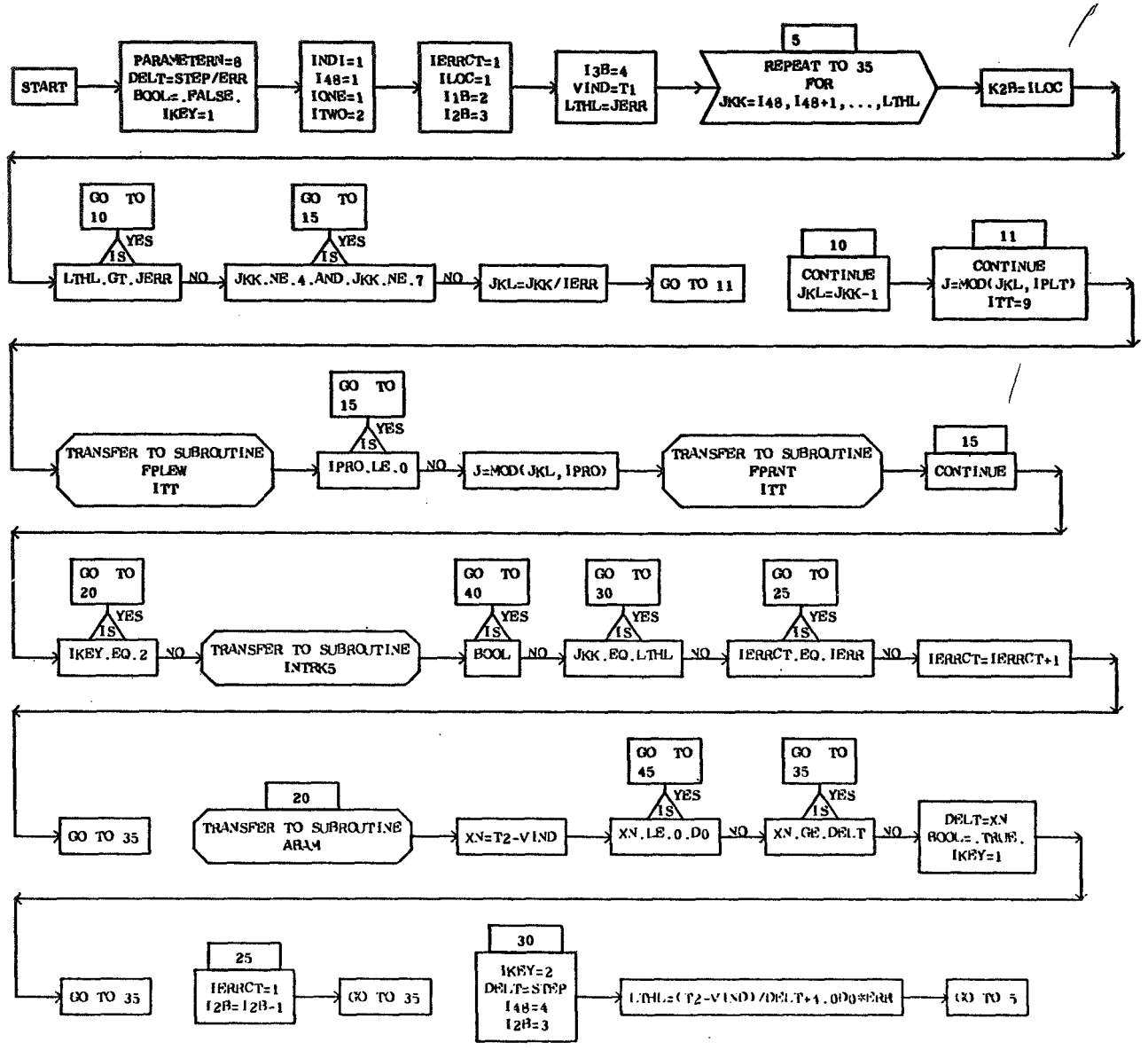

COMPUTED GO TO
IF THE VALUE OF KEY IS
TRANSFER TO STATEMENT
1 310
2 320
3 330
310
B1 WRITE
310
WRITE(6,140)
GO TO 340
320
A2 READ
320
READ(5,210)C,DEL
B1 WRITE
WRITE(6,150)C,DEL
GO TO 340
330
B1 WRITE
330
WRITE(6,160)
A2 READ
READ(5,240)FL1,FL2,IFL
COMPUTED GO TO
IF THE VALUE OF IFL IS
TRANSFER TO STATEMENT
1 331
2 332
3 333
4 334
331
B1 WRITE
331
WRITE(6,161)FL1,FL2
GO TO 340
332
B1 WRITE
332
WRITE(6,162)FL1,FL2
GO TO 340
333
B1 WRITE
333
WRITE(6,163)FL1,FL2
GO TO 340
334
B1 WRITE
334
WRITE(6,164)FL1,FL2
340
CONTINUE
GO TO 345
YES
IS
IHMAX.EQ.0
NO
A2 READ
READ(5,210)HMAX
B1 WRITE
WRITE(6,170)HMAX
345
CONTINUE
NOK=KON

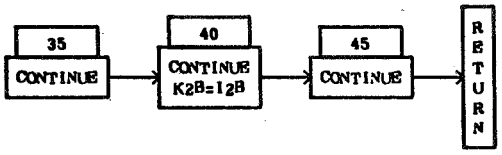

GO TO 400
YES
IS
NOK.LE.0
NO
B1 WRITE
WRITE(6,130)NOK
REPEAT TO 350 FOR I=1,1+1,...,KON
A2 READ
READ(5,230)PLACE,CON(I)
B1 WRITE
WRITE(6,135)PLACE,CON(I)
350
CONTINUE
400
CONTINUE
NO
IS
SWCH.EQ.0
YES
B1 WRITE
WRITE(6,165)
RETURN

SUBROUTINE F7

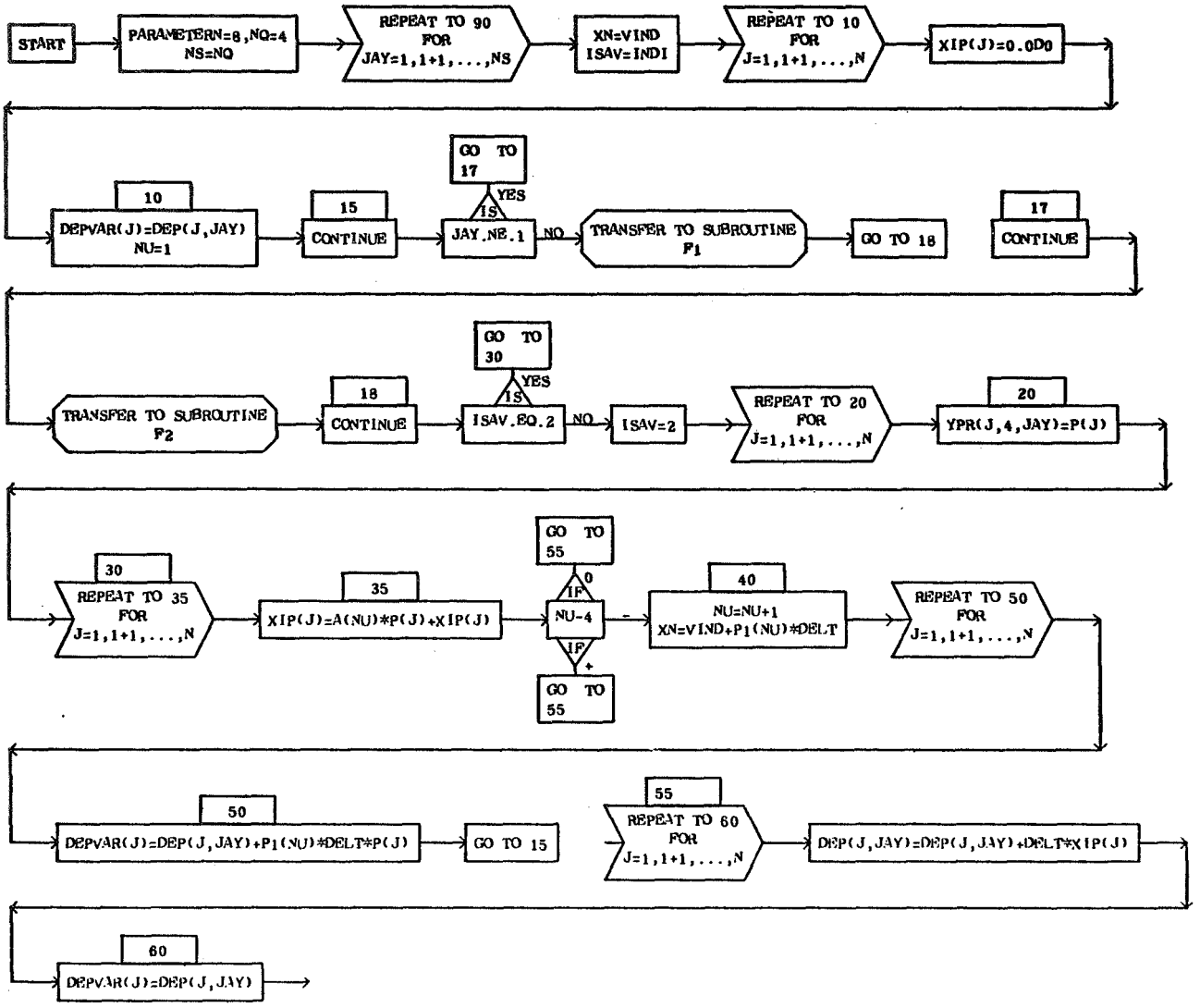


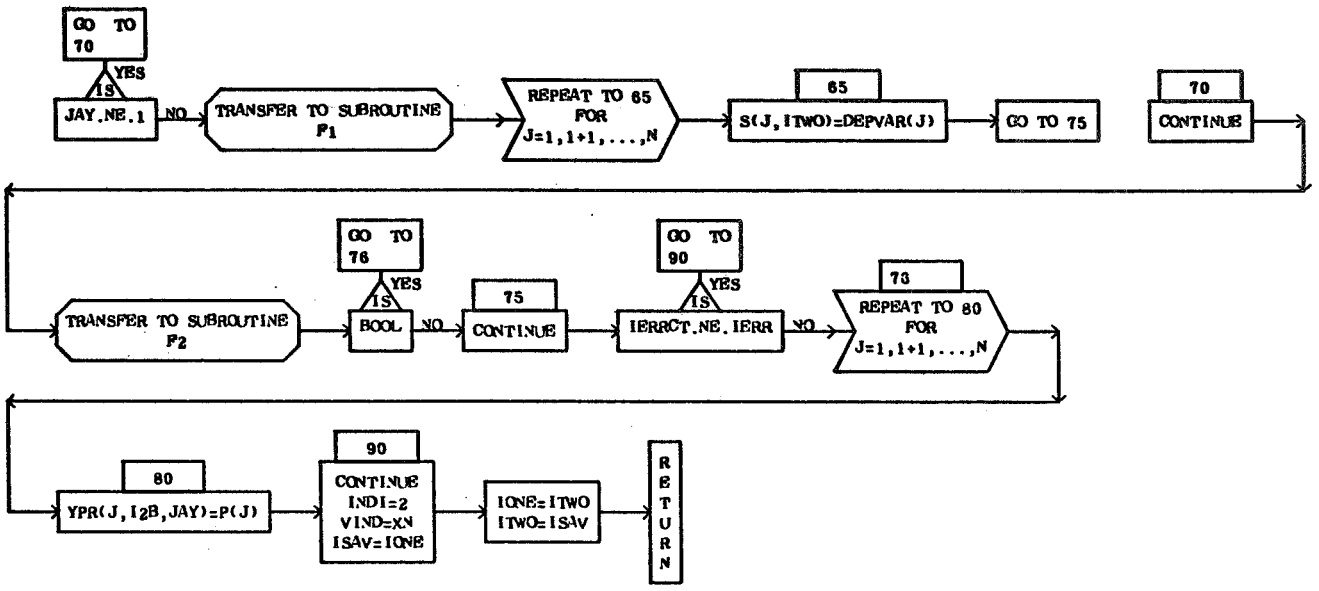
SUBROUTINE INTEG



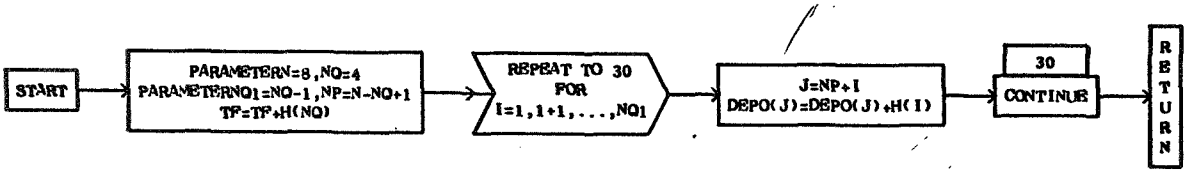

35
CONTINUE
40
CONTINUE
K2B=I2B
45
CONTINUE
RETURN

SUBROUTINE INTRK5

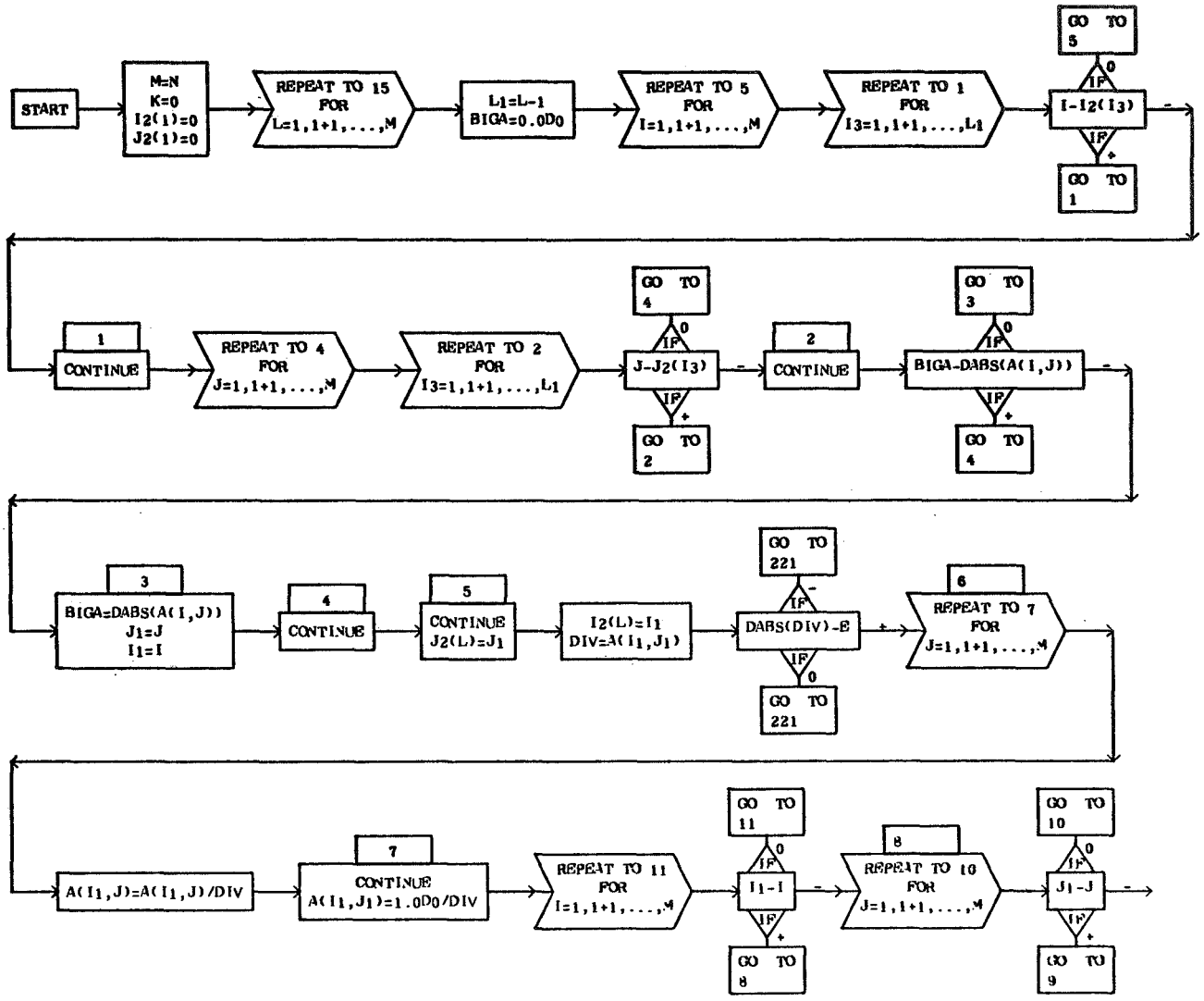


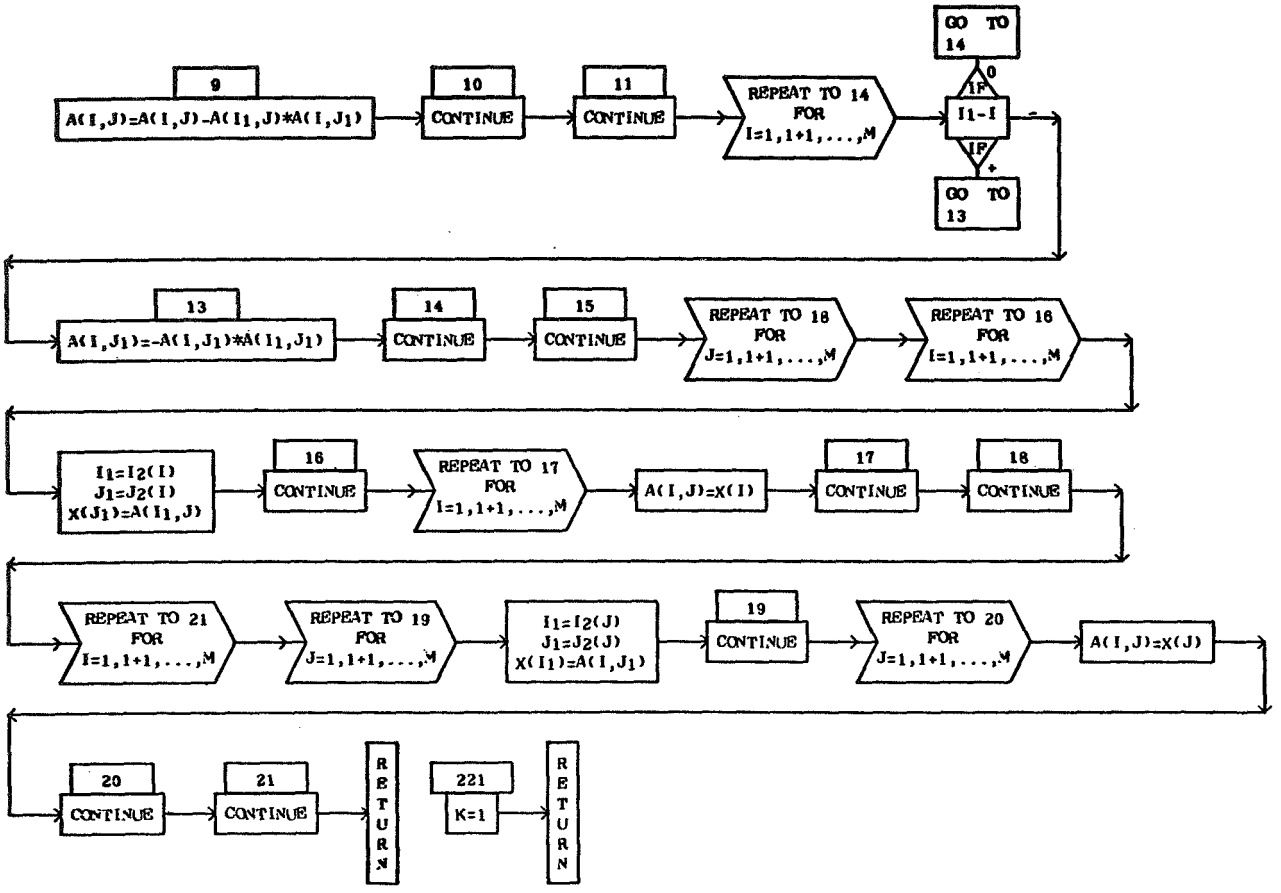

GO TO 70
IS
YES
JAY.NE.1
NO
TRANSFER TO SUBROUTINE P1
REPEAT TO 65 FOR J=1,1+1,...,N
65
S(J,ITWO)=DEPVAR(J)
GO TO 75
70
CONTINUE
TRANSFER TO SUBROUTINE P2
GO TO 76
IS
YES
BOOL
NO
75
CONTINUE
GO TO 90
IS
YES
IERRCT.NE.IERR
NO
73
REPEAT TO 80 FOR J=1,1+1,...,N
80
YPR(J,I2B,JAY)=P(J)
90
CONTINUE
INDI=2
VIND=XN
ISAV=IONE
IONE=ITWO
ITWO=ISAV
RETURN

SUBROUTINE ITER

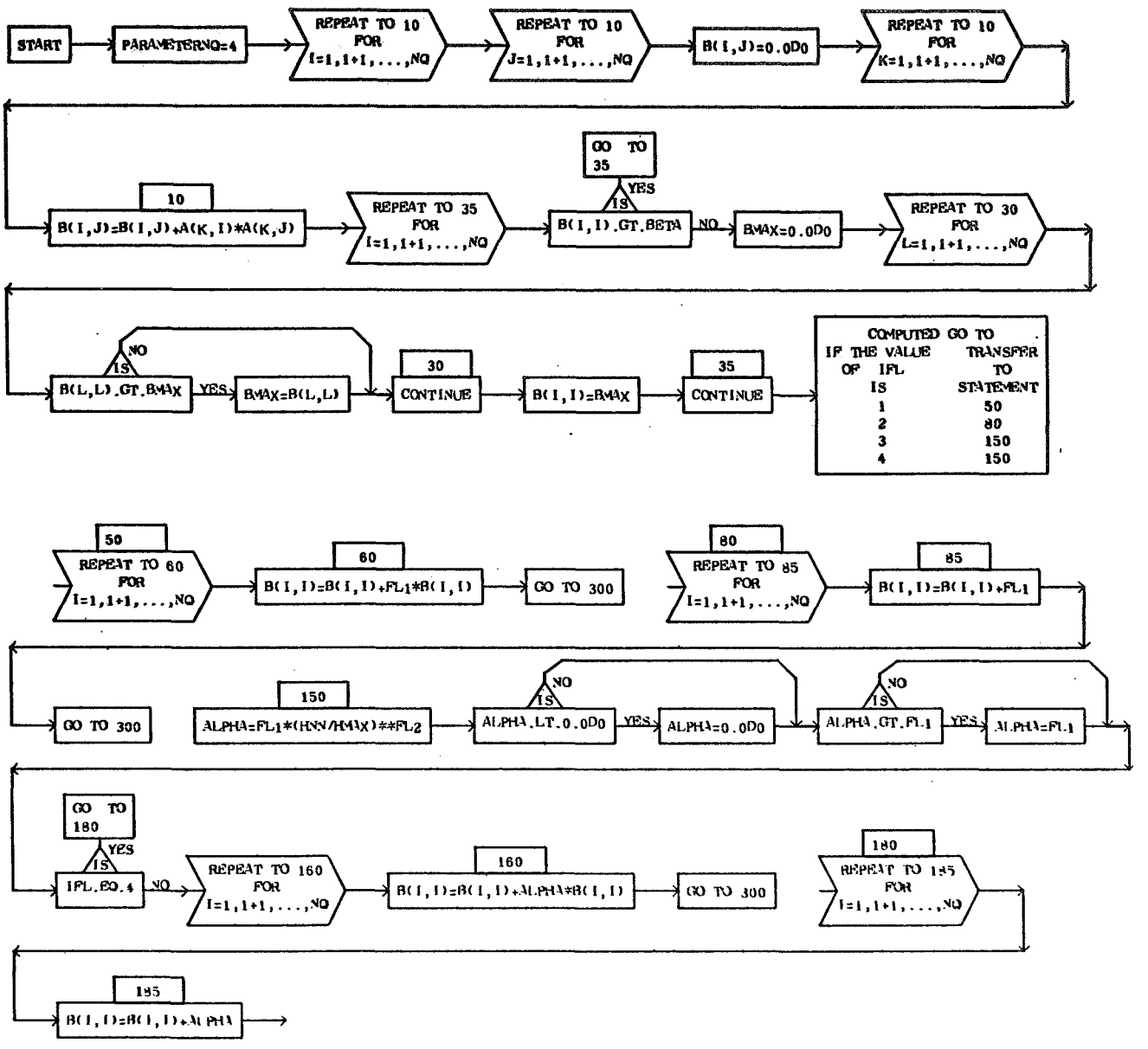


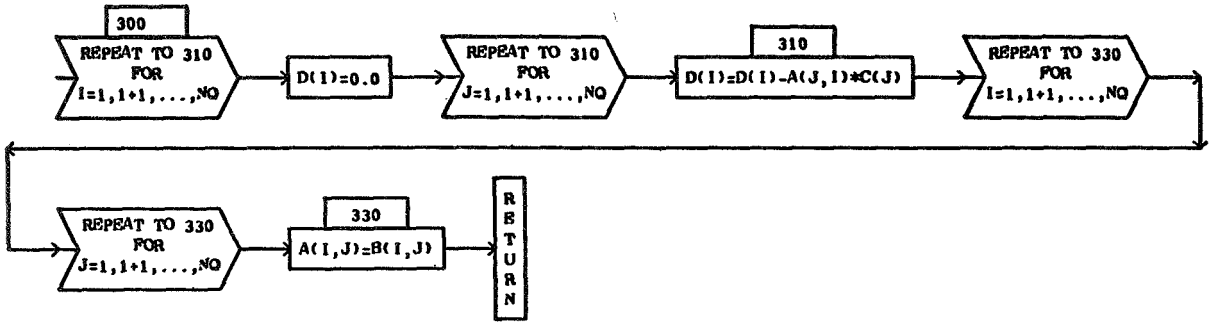
SUBROUTINE MINVDP(A,N,K,X,J2,I2)




9
A(I,J)=A(I,J)-A(I1,J)*A(I,J1)
10
CONTINUE
11
CONTINUE
REPEAT TO 14
FOR
I=1,1+1,...,M
GO TO
14
0
IF
I1-I
IF
+
GO TO
13
13
A(I,J1)=-A(I,J1)*A(I1,J1)
14
CONTINUE
15
CONTINUE
REPEAT TO 18
FOR
J=1,1+1,...,M
REPEAT TO 16
FOR
I=1,1+1,...,M
I1=I2(I)
J1=J2(I)
X(J1)=A(I1,J)
16
CONTINUE
REPEAT TO 17
FOR
I=1,1+1,...,M
A(I,J)=X(I)
17
CONTINUE
18
CONTINUE
REPEAT TO 21
FOR
I=1,1+1,...,M
REPEAT TO 19
FOR
J=1,1+1,...,M
I1=I2(J)
J1=J2(J)
X(I1)=A(I,J1)
19
CONTINUE
REPEAT TO 20
FOR
J=1,1+1,...,M
A(I,J)=X(J)
20
CONTINUE
21
CONTINUE
RETURN
221
K=1
RETURN

SUBROUTINE MLTPLY




300
REPEAT TO 310
FOR
I=1,1+1,...,NQ
D(I)=0.0
REPEAT TO 310
FOR
J=1,1+1,...,NQ
310
D(I)=D(I)-A(J,I)*C(J)
REPEAT TO 330
FOR
I=1,1+1,...,NQ
REPEAT TO 330
FOR
J=1,1+1,...,NQ
330
A(I,J)=B(I,J)
RETURN

4. SAMPLE INPUT

```
          TIME IN DAYS
          CONTROL ANGLE IN DEGREES
    0.0                          1.0
    1.0                          0.0
   -1.0                          0.0
   -0.49484461                  -1.0785625
    0.0                          0.81012728
    1.523679                     0.0
    0.0                          3.3194865
    0.3                  -001    0.1                  -005
     2    25     5     5         1     1
GRAVITATIONAL CONSTANT           1.0
INITIAL MASS                     1.0
MASS FLOW RATE                   0.74800391           -001
THRUST                           0.14012969
    0.0                          1.0
    1.0                          0.0
   -1.0                          0.0
   -0.49484461                  -1.0785625
    0.0                          0.81012728
    1.523679                     0.0
    0.0                          3.3194865
    0.3                  -001    0.1                  -005
     2    25     5     5         2     1
    0.5                          0.1
GRAVITATIONAL CONSTANT           1.0
INITIAL MASS                     1.0
MASS FLOW RATE                   0.74800391           -001
THRUST                           0.14012969
    0.0                          1.0
    1.0                          0.0
   -1.0                          0.0
   -0.49484461                  -1.0785625
    0.0                          0.81012728
    1.523679                     0.0
    0.0                          3.3194865
    0.3                  -001    0.1                  -005
     2    25     5     5         3     1
    2.0                          2.0                          3
GRAVITATIONAL CONSTANT           1.0
INITIAL MASS                     1.0
MASS FLOW RATE                   0.74800391           -001
THRUST                           0.14012969
    0.0                          1.0
    1.0                          0.0
   -1.0                          0.0
   -0.395874528                 -0.86285152
    0.0                          0.81012728
    1.523679                     0.0
    0.0                          3.3194865
    0.3                  -001    0.1                  -005
     2    25     5     5         3     1
    2.0                          2.0                          3
GRAVITATIONAL CONSTANT           1.0
INITIAL MASS                     1.0
MASS FLOW RATE                   0.74800391           -001
THRUST                           0.14012969
```

```
    0.0                                 0.81012728
    1.523679                            0.0
    0.0                                 3.3194865
    0.3                        -001     0.1                 -005
     2    25     5     5                3
    2.0                                 2.0                         3
GRAVITATIONAL CONSTANT                  1.0
INITIAL MASS                            1.0
MASS FLOW RATE                          0.74800391          -001
THRUST                                  0.14012969
' EOF
```

5. SAMPLE OUTPUT

```
                    METHOD OF PERTURBATION FUNCTIONS

                    TWO-POINT BOUNDARY VALUE PROBLEM


                 INITIAL VALUE OF THE DEPENDENT VARIABLES

  1        .0000000000000000          2        .1000000000000000+001
  3        .1000000000000000+001      4        .0000000000000000
  5       -.1000000000000000+001      6        .0000000000000000
  7       -.4948446099999999+000      8       -.1078562500000000+001

                 DESIRED VALUES OF THE TERMINAL VARIABLES

  1        .0000000000000000          2        .8101272799999999+000
  3        .1523679000000000+001      4        .0000000000000000

                      THE INITIAL TIME INTERVAL
FROM       .0000000000000000      /   TO       .3319486499999999+001

      THE INTEGRATION STEP SIZE IS     .3000000000000000-001

        THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS    25

THE ACCURACY REQUIRED FOR TERMINATION IS     .1000000000000000-005

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS     2

   EVERY    5TH POINT WILL BE PLOTTED FOR EACH    5TH ITERATION

       EVERY    0TH POINT WILL BE PRINTED ON EACH ITERATION

                     PRINT CONTROL SWITCH =  1

                     PUNCH CONTROL SWITCH =  0

                    NORMAL CORRECTION PROCEDURE

       4 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

   GRAVITATIONAL CONSTANT                    .1000000000000000+001
   INITIAL MASS                              .1000000000000000+001
   MASS FLOW RATE                            .7480039100000000-001
   THRUST                                    .1401296900000000+000
```

```
                    BEGINNING THE   1 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS        .1170
                    ELAPSED TIME WAS        .1170

  THE DEPENDENT VARIABLES AT TIME =     .0000000000000000

1  1     .0000000000000000          1  2     .0000000000000000
2  1     .1000000000000000+001      2  2     .0000000000000000
3  1     .1000000000000000+001      3  2     .0000000000000000
4  1     .0000000000000000          4  2     .0000000000000000
5  1    -.1000000000000000+001      5  2     .0000000000000000
6  1     .0000000000000000          6  2     .1000000000000000+001
7  1    -.4948446099999999+000      7  2     .0000000000000000
8  1    -.1078562500000000+001      8  2     .0000000000000000

1  3     .0000000000000000          1  4     .0000000000000000
2  3     .0000000000000000          2  4     .0000000000000000
3  3     .0000000000000000          3  4     .0000000000000000
4  3     .0000000000000000          4  4     .0000000000000000
5  3     .0000000000000000          5  4     .0000000000000000
6  3     .0000000000000000          6  4     .0000000000000000
7  3     .1000000000000000+001      7  4     .0000000000000000
8  3     .0000000000000000          8  4     .1000000000000000+001

  THE DEPENDENT VARIABLES AT TIME =     .3319486499999999+001

1  1     .1340051529213354-006      1  2    -.1394670731000021+001
2  1     .8101273411074560+000      2  2    -.1863971775431036+001
3  1     .1523679074695318+001      3  2     .2795203649501503+000
4  1     .2490447285702952+001      4  2    -.1981448782912016+001
5  1    -.7118565185737861+000      5  2     .4169080799384614+001
6  1     .0000000000000000          6  2     .1000000000000000+001
7  1     .7584227019400643+000      7  2    -.3860097934044046+001
8  1    -.6742203751794457+000      8  2     .7973314319544971+001

1  3     .5272719904312362+001      1  4     .2390304718702410+001
2  3     .6285599236809059+000      2  4     .1344724990351260+001
3  3     .4832814651829563+001      3  4     .1272919107827973+001
4  3     .1493460241952366+001      4  4     .1455845270113698+001
5  3     .3005228841106784+001      5  4     .2039032539623578+000
6  3     .0000000000000000          6  4     .0000000000000000
7  3    -.3009761305147651+001      7  4    -.5954886874401262+000
8  3    -.1533775869407431+001      8  4    -.3954961594593700+001

   TIME AT COMPLETION OF FORWARD INTEGRATION IS        2.5220
                    ELAPSED TIME WAS        2.4050

                    THE TERMINAL CONSTRAINT VECTOR

1       .1340051529213354-006      2       .6110745605792528-007
3       .7469531817377861-007      4       .0000000000000000

THE NORM OF THE TERMINAL CONSTRAINTS      .1651390103895012-006
```

THE CORRECTIONS AT THE 1 TH ITERATION

1	.4263213842357138-025	2	-.3051449300367758-008
3	-.4709508566341746-007	4	.3835757786522355-007

********** CONVERGENCE HAS BEEN ACHEIVED **********

THE FINAL TIME IS 2.6470

METHOD OF PERTURBATION FUNCTIONS

TWO-POINT BOUNDARY VALUE PROBLEM

INITIAL VALUE OF THE DEPENDENT VARIABLES

1	.0000000000000000	2	.1000000000000000+001
3	.1000000000000000+001	4	.0000000000000000
5	-.1000000000000000+001	6	.0000000000000000
7	-.4948446099999999+000	8	-.1078562500000000+001

DESIRED VALUES OF THE TERMINAL VARIABLES

1	.0000000000000000	2	.8101272799999999+000
3	.1523679000000000+001	4	.0000000000000000

THE INITIAL TIME INTERVAL

FROM .0000000000000000 / TO .3319486499999999+001

THE INTEGRATION STEP SIZE IS .3000000000000000-001

THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS 25

THE ACCURACY REQUIRED FOR TERMINATION IS .1000000000000000-005

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS 2

EVERY 5TH POINT WILL BE PLOTTED FOR EACH 5TH ITERATION

EVERY 0TH POINT WILL BE PRINTED ON EACH ITERATION

PRINT CONTROL SWITCH = 1

PUNCH CONTROL SWITCH = 0

FRACTIONAL CORRECTION PROCEDURE

C = .5000000000000000+000 DEL = .1000000000000000+000

4 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

GRAVITATIONAL CONSTANT	.1000000000000000+001
INITIAL MASS	.1000000000000000+001
MASS FLOW RATE	.7480039100000000-001
THRUST	.1401296900000000+000

BEGINNING THE 1 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS .1170
ELAPSED TIME WAS .1170

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	-.4948446099999999+000	7	2	.0000000000000000
8	1	-.1078562500000000+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3319486499999999+001

1	1	.1340051529213354-006	1	2	-.1394670731000021+001
2	1	.8101273411074560+000	2	2	-.1863971775431036+001
3	1	.1523679074695318+001	3	2	.2795203649501503+000
4	1	.2490447285702952+001	4	2	-.1981448782912016+001
5	1	-.7118565185737861+000	5	2	.4169080799384614+001
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	.7584227019400643+000	7	2	-.3860097934044046+001
8	1	-.6742203751794457+000	8	2	.7973314319544971+001

1	3	.5272719904312362+001	1	4	.2390304718702410+001
2	3	.8285599236809059+000	2	4	.1344724990351260+001
3	3	.4832814651829563+001	3	4	.1272919107827973+001
4	3	.1493460241952366+001	4	4	.1455845270113698+001
5	3	.3005228841106784+001	5	4	.2039032539623578+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3009761305147651+001	7	4	-.5954886874401262+000
8	3	-.1533775869407431+001	8	4	-.3954961594593700+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 2.5160
ELAPSED TIME WAS 2.3990

THE TERMINAL CONSTRAINT VECTOR

1	.1340051529213354-006	2	.6110745605792528-007
3	.7469531817377861-007	4	.0000000000000000

THE NORM OF THE TERMINAL CONSTRAINTS .1651390103895012-006

THE A MATRIX

1	1	-.1394670731000021+001	1	2	.5272719904312362+001
2	1	-.1863971775431036+001	2	2	.8285599236809059+000
3	1	.2795203649501503+000	3	2	.4832814651829563+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2390304718702410+001	1	4	-.1393234470670404+000
2	3	.1344724990351260+001	2	4	.1238553480234650+000
3	3	.1272919107827973+001	3	4	.1340051529213354-006
4	3	.0000000000000000	4	4	.0000000000000000

THE FRACTIONAL CORRECTION CONSTANT .5000000000000000+000

THIS IS THE 1 TH ATTEMPT TO CORRECT FROM THE 1 TH ITERATION

THE CORRECTIONS AT THE 1 TH ITERATION

1	.2131606921178569-025	2	-.1525724650183879-008
3	-.2354754283170873-007	4	.1917878893261177-007

********** CONVERGENCE HAS BEEN ACHEIVED **********

THE FINAL TIME IS 2.6840

METHOD OF PERTURBATION FUNCTIONS

TWO-POINT BOUNDARY VALUE PROBLEM

INITIAL VALUE OF THE DEPENDENT VARIABLES

1	.0000000000000000	2	.1000000000000000+001
3	.1000000000000000+001	4	.0000000000000000
5	-.1000000000000000+001	6	.0000000000000000
7	-.4948446099999999+000	8	-.1078562500000000+001

DESIRED VALUES OF THE TERMINAL VARIABLES

1	.0000000000000000	2	.8101272799999999+000
3	.1523679000000000+001	4	.0000000000000000

THE INITIAL TIME INTERVAL

FROM .0000000000000000 / TO .3319486499999999+001

THE INTEGRATION STEP SIZE IS .3000000000000000-001

THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS 25

THE ACCURACY REQUIRED FOR TERMINATION IS .1000000000000000-005

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS 2

EVERY 5TH POINT WILL BE PLOTTED FOR EACH 5TH ITERATION

EVERY 0TH POINT WILL BE PRINTED ON EACH ITERATION

PRINT CONTROL SWITCH = 1

PUNCH CONTROL SWITCH = 0

MINIMUM NORM CORRECTION PROCEDURE

VARIABLE ALPHA PROCEDURE

ALPHA = .2000000000000000+001 P = .2000000000000000+001

4 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

GRAVITATIONAL CONSTANT	.1000000000000000+001
INITIAL MASS	.1000000000000000+001
MASS FLOW RATE	.7480039100000000-001
THRUST	.1401296900000000+000

BEGINNING THE 1 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS .1230
ELAPSED TIME WAS .1230

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	-.4948446099999999+000	7	2	.0000000000000000
8	1	-.1078562500000000+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3319486499999999+001

1	1	.1340051529213354-006	1	2	-.1394670731000021+001
2	1	.8101273411074560+000	2	2	-.1863971775431036+001
3	1	.1523679074695318+001	3	2	.2795203649501503+000
4	1	.2490447285702952+001	4	2	-.1981448782912016+001
5	1	-.7118565185737861+000	5	2	.4169080799384614+001
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	.7584227019400643+000	7	2	-.3860097934044046+001
8	1	-.6742203751794457+000	8	2	.7973314319544971+001

1	3	.5272719904312362+001	1	4	.2390304718702410+001
2	3	.8285599236809059+000	2	4	.1344724990351260+001
3	3	.4832814651829563+001	3	4	.1272919107827973+001
4	3	.1493460241952366+001	4	4	.1455845270113698+001
5	3	.3005228841106784+001	5	4	.2039032539623578+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3009761305147651+001	7	4	-.5954886874401262+000
8	3	-.1533775869407431+001	8	4	-.3954961594593700+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 2.5240
ELAPSED TIME WAS 2.4010

THE TERMINAL CONSTRAINT VECTOR

1	.1340051529213354-006	2	.6110745605792528-007
3	.7469531817377861-007	4	.0000000000000000

THE NORM OF THE TERMINAL CONSTRAINTS .1651390103895012-006

THE A MATRIX

1	1	-.1394670731000021+001	1	2	.5272719904312362+001
2	1	-.1863971775431036+001	2	2	.8285599236809059+000
3	1	.2795203649501503+000	3	2	.4832814651829563+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2390304718702410+001	1	4	-.1393234470670404+000
2	3	.1344724990351260+001	2	4	.1238553480234650+000
3	3	.1272919107827973+001	3	4	.1340051529213354-006
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.1949288658580058+002	1	2	-.7547250320084354+001
2	1	-.7547250320084354+001	2	2	.1555325525862011+003
3	1	-.5484410643504918+001	3	2	.1986937461795605+002
4	1	-.3655250172832569-001	4	2	-.6319912871599335+000
1	3	-.5484410643504918+001	1	4	-.3655250172832569-001
2	3	.1986937461795605+002	2	4	-.6319912871599335+000
3	3	.2742649500899931+002	3	4	-.1664739406967066+000
4	3	-.1664739406967066+000	4	4	.1042535104100221+000

THE CORRECTIONS AT THE 1 TH ITERATION

1	.9079297568059467-008	2	-.4921214232811461-008
3	-.1239660938720785-007	4	.6004170665702625-007

********** CONVERGENCE HAS BEEN ACHEIVED **********

THE FINAL TIME IS 2.7360

METHOD OF PERTURBATION FUNCTIONS

TWO-POINT BOUNDARY VALUE PROBLEM

INITIAL VALUE OF THE DEPENDENT VARIABLES

1	.0000000000000000	2	.1000000000000000+001
3	.1000000000000000+001	4	.0000000000000000
5	-.1000000000000000+001	6	.0000000000000000
7	-.3958745279999999+000	8	-.8628515199999999+000

DESIRED VALUES OF THE TERMINAL VARIABLES

1	.0000000000000000	2	.8101272799999999+000
3	.1523679000000000+001	4	.0000000000000000

THE INITIAL TIME INTERVAL

FROM .0000000000000000 / TO .3319486499999999+001

THE INTEGRATION STEP SIZE IS .3000000000000000-001

THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS 25

THE ACCURACY REQUIRED FOR TERMINATION IS .1000000000000000-005

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS 2

EVERY 5TH POINT WILL BE PLOTTED FOR EACH 5TH ITERATION

EVERY 0TH POINT WILL BE PRINTED ON EACH ITERATION

PRINT CONTROL SWITCH = 1

PUNCH CONTROL SWITCH = 0

MINIMUM NORM CORRECTION PROCEDURE

VARIABLE ALPHA PROCEDURE

ALPHA = .2000000000000000+001 P = .2000000000000000+001

4 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

GRAVITATIONAL CONSTANT	.1000000000000000+001
INITIAL MASS	.1000000000000000+001
MASS FLOW RATE	.7480039100000000-001
THRUST	.1401296900000000+000

BEGINNING THE 1 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS .1240
ELAPSED TIME WAS .1240

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	-.3958745279999999+000	7	2	.0000000000000000
8	1	-.8628515199999999+000	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3319486499999999+001

1	1	.4393133799683166+000	1	2	.5699911572726967+000
2	1	.9030159840681569+000	2	2	-.2769863623328869+000
3	1	.1826985034993032+001	3	2	.1035177230850690+001
4	1	.2695180469031795+001	4	2	-.6300583362142748+000
5	1	-.5862242216022068+000	5	2	.3273930891986523+001
6	1	.0000000000000000	6	2	.1000000000000000+001
7	1	.3672794036554746+000	7	2	-.2814215594421436+001
8	1	-.1435088556299252+001	8	2	.7048535636381724+001

1	3	.9208285722259291+000	1	4	.1337858902226654+000
2	3	-.1292596787507494+000	2	4	.1341257999600308+000
3	3	.8933933781390381+000	3	4	-.2445348688635943+000
4	3	.2245413927907733+000	4	4	.3775375111605853+000
5	3	.7301358683896568+000	5	4	-.2421948794618864+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.2067285202548250+001	7	4	-.6984550113388558+000
8	3	-.2502866970053362+000	8	4	-.2868997681268256+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 2.5240
ELAPSED TIME WAS 2.4000

THE TERMINAL CONSTRAINT VECTOR

1	.4393133799683166+000	2	.9288870406815695-001
3	.3033060349930329+000	4	.0000000000000000

THE NORM OF THE TERMINAL CONSTRAINTS .5418663193314779+000

THE A MATRIX

1	1	.5699911572726967+000	1	2	.9208285722259291+000
2	1	-.2769863623328869+000	2	2	-.1292596787507494+000
3	1	.1035177230850690+001	3	2	.8933933781390381+000
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.1337858902226654+000	1	4	.1005183228839424+000
2	3	.1341257999600308+000	2	4	-.3654140383864541-001
3	3	-.2445348688635943+000	3	4	.4393133799683166+000
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.7419609790677531+001	1	2	.1485487794988623+001
2	1	.1485487794988623+001	2	2	.4988355156243217+001
3	1	-.2140311714277947+000	3	2	-.1126090201041422+000
4	1	.5221832338627916+000	4	2	.4897631384566236+000
1	3	-.2140311714277947+000	1	4	.5221832338627916+000
2	3	-.1126090201041422+000	2	4	.4897631384566236+000
3	3	.2870570901831736+000	3	4	-.9888065147137800-001
4	3	-.9888065147137800-001	4	4	.6133063597472582+000

THE CORRECTIONS AT THE 1 TH ITERATION

1	-.4221201814654489-001	2	-.1052090398174694+000
3	-.1259232058014515+000	4	-.1840721347124976+000

BEGINNING THE 2 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 2.7350
ELAPSED TIME WAS .2110

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.4221201814654489-001	6	2	.1000000000000000+001
7	1	-.5010835678174694+000	7	2	.0000000000000000
8	1	-.9887747258014515+000	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3135414365287502+001

1	1	.1994873188021790+000	1	2	.1238421621978655+000
2	1	.9020494976662636+000	2	2	-.5080784428607843+000
3	1	.1574107834433610+001	3	2	.8475395073996991+000
4	1	.2535181753024635+001	4	2	-.8309029900726967+000
5	1	-.8326715153627686+000	5	2	.4002854537931061+001
6	1	-.4221201814654489-001	6	2	.1000000000000000+001
7	1	.7990421600745388+000	7	2	-.3324284777156963+001
8	1	-.1125205657676754+001	8	2	.6745501161216616+001

1	3	.3032372083904383+001	1	4	.8555863350487180+000
2	3	-.1146433650501800-001	2	4	.3660847515889093+000
3	3	.2893751399111466+001	3	4	.2953189656545785+000
4	3	.6261444967761286+000	4	4	.6114530187334158+000
5	3	.2249893120068476+001	5	4	-.1808538684731704+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.2719017398540971+001	7	4	-.5020126380482614+000
8	3	-.9472558769528791+000	8	4	-.3216695622133141+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 5.0620
ELAPSED TIME WAS 2.3270

THE TERMINAL CONSTRAINT VECTOR

1	.1994873188021790+000	2	.9192221766626364-001
3	.5042883443361056-001	4	-.4221201814654489-001

THE NORM OF THE TERMINAL CONSTRAINTS .2292810639409694+000

THE A MATRIX

1	1	.1238421621978655+000	1	2	.3032372083904383+001
2	1	-.5080784428607843+000	2	2	-.1148433650501800-001
3	1	.8475395073996991+000	3	2	.2893751399111466+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.8555863350487180+000	1	4	.7349727471776571-002
2	3	.3660847515889093+000	2	4	.3494073635183525-001
3	3	.2953189656545785+000	3	4	.1994873188021790+000
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.2705032179369647+001	1	2	.2833939094608047+001
2	1	.2833939094608047+001	2	2	.2386042090763186+002
3	1	.1702523818136807+000	3	2	.3444831547342279+001
4	1	.1522309551337057+000	4	2	.5991525451245235+000
1	3	.1702523818136807+000	1	4	.1522309551337057+000
2	3	.3444831547342279+001	2	4	.5991525451245235+000
3	3	.1294603974565704+001	3	4	.7799198582874405-001
4	3	.7799198582874405-001	4	4	.5577649987030418-001

THE CORRECTIONS AT THE 2 TH ITERATION

1	.3340495839818038-001	2	-.1692635058238253-001
3	-.1291077692269256+000	4	.6950028160102317-002

BEGINNING THE 3 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 5.2720
ELAPSED TIME WAS .2100

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.8807059748364503-002	6	2	.1000000000000000+001
7	1	-.5180099183998519+000	7	2	.0000000000000000
8	1	-.1117882495028377+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3142364393447604+001

1	1	-.9231252372457258-001	1	2	-.1315202793544773-002
2	1	.7314536006146396+000	2	2	-.2156658372477752+001
3	1	.1438023206640970+001	3	2	.1248192371447132+001
4	1	.2342897200439062+001	4	2	-.1493078999043401+001
5	1	-.7278964344283806+000	5	2	.5136398369717176+001
6	1	-.8807059748364503-002	6	2	.1000000000000000+001
7	1	.7665687371058279+000	7	2	-.4824380985158326+001
8	1	-.4156486732513994+000	8	2	.6792843686655758+001

1	3	.2508134205968545+001	1	4	.6042705845667370+000
2	3	.7048112314969139+000	2	4	.1377479392436861+001
3	3	.2229557687920515+001	3	4	-.1614446324458156+000
4	3	.7633571644328466+000	4	4	.9159246240451239+000
5	3	.1183326975107175+001	5	4	-.9616730132444799+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.1576450945521766+001	7	4	.5337697539689575+000
8	3	-.1588649034731688+001	8	4	-.3584841440136115+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 7.6000
ELAPSED TIME WAS 2.3280

THE TERMINAL CONSTRAINT VECTOR

1	-.9231252372457258-001	2	-.7867367938536031-001
3	-.8565579335902925-001	4	-.8807059748364503-002

THE NORM OF THE TERMINAL CONSTRAINTS .1487468624939908+000

THE A MATRIX

1	1	-.1315202793544773-002	1	2	.2508134205968545+001
2	1	-.2156658372477752+001	2	2	.7048112314969139+000
3	1	.1248192371447132+001	3	2	.2229557687920515+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.6042705845667370+000	1	4	-.2725628931284872+000
2	3	.1377479392436861+001	2	4	.1342731987234617+000
3	3	-.1614446324458156+000	3	4	-.9231252372457258-001
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.8295652050709099+001	1	2	.1259581149225245+001
2	1	.1259581149225245+001	2	2	.1353053245803726+002
3	1	-.3173061161605356+001	3	2	.2126504548314416+001
4	1	-.4044467306499074+000	4	2	-.7948031539454608+000
1	3	-.3173061161605356+001	1	4	-.4044467306499074+000
2	3	.2126504548314416+001	2	4	-.7948031539454608+000
3	3	.2633579644812991+001	3	4	.3516018689905522-001
4	3	.3516018689905522-001	4	4	.1160392091104279+000

THE CORRECTIONS AT THE 3 TH ITERATION

1	.3977544980233116-002	2	.2975532391312685-001
3	.3768163143842494-001	4	.1231594694984688-001

BEGINNING THE 4 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 7.8080
ELAPSED TIME WAS .2080

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.4829514768131386-002	6	2	.1000000000000000+001
7	1	-.4882545944867250+000	7	2	.0000000000000000
8	1	-.1080200863589952+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3154680340397451+001

1	1	.6284025235808489-001	1	2	-.1522378532924491+001
2	1	.8065984615459050+000	2	2	-.1859759822469675+001
3	1	.1547885248260215+001	3	2	.5999994284889531-001
4	1	.2422309454630739+001	4	2	-.1938868204806252+001
5	1	-.6799926099886249+000	5	2	.3670345759321230+001
6	1	-.4829514768131386-002	6	2	.1000000000000000+001
7	1	.6976232172234834+000	7	2	-.3598155401678205+001
8	1	-.5879696425191652+000	8	2	.7106255401576260+001

1	3	.6126851133969627+001	1	4	.2765968068571844+001
2	3	.1422977616154302+001	2	4	.1522782491203944+001
3	3	.5040184459718744+001	3	4	.1509379884454352+001
4	3	.2133002943092935+001	4	4	.1630356379138410+001
5	3	.2850514465236823+001	5	4	.3152149329089573+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3047106528487672+001	7	4	-.5566577814574003+000
8	3	-.2091187503760970+001	8	4	-.3856906853806662+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 10.1470
ELAPSED TIME WAS 2.3390

THE TERMINAL CONSTRAINT VECTOR

1	.6284025235808489-001	2	-.3528818454094940-002
3	.2420624826021514-001	4	-.4829514768131386-002

THE NORM OF THE TERMINAL CONSTRAINTS .6760633508659172-001

THE A MATRIX

1	1	-.1522378532924491+001	1	2	.6126851133969627+001
2	1	-.1859759822469675+001	2	2	.1422977616154302+001
3	1	.5999994284889531-001	3	2	.5040184459718744+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2765968068571844+001	1	4	-.1372973751262878+000
2	3	.1522782491203944+001	2	4	.8545288415985267-001
3	3	.1509379884454352+001	3	4	.6284025235808489-001
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.6991022708145193+001	1	2	-.1167137246004586+002
2	1	-.1167137246004586+002	2	2	.6698923323225061+002
3	1	-.6952297399045237+001	3	2	.2672111303414698+002
4	1	.5386714739458759-001	4	2	-.4028765737082296+000
1	3	-.6952297399045237+001	1	4	.5386714739458759-001
2	3	.2672111303414698+002	2	4	-.4028765737082296+000
3	3	.1262897988452051+002	3	4	-.1547841868332113+000
4	3	-.1547841868332113+000	4	4	.3103881589858251-001

THE CORRECTIONS AT THE 4 TH ITERATION

1	.4164955715104852-003	2	-.5544137570013910-002
3	-.2381941291416366-002	4	.1541133933505249+000

BEGINNING THE 5 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 10.3560
ELAPSED TIME WAS .2090

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.4413019196620901-002	6	2	.1000000000000000+001
7	1	-.4937987320567389+000	7	2	.0000000000000000
8	1	-.1082582804881368+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3308793733747976+001

1	1	.3543537076885874-002	1	2	-.1391361569232885+001
2	1	.8124370545538605+000	2	2	-.1810006433295714+001
3	1	.1522414831387861+001	3	2	.2458130897395349+000
4	1	.2489156105214893+001	4	2	-.1943279055614225+001
5	1	-.7255672257196072+000	5	2	.4136001262217634+001
6	1	-.4413019196620901-002	6	2	.1000000000000000+001
7	1	.7708253439118592+000	7	2	-.3822817950986701+001
8	1	-.6863372913144886+000	8	2	.7928616315944276+001

1	3	.5294724581476790+001	1	4	.2390195774391711+001
2	3	.8260307928105425+000	2	4	.1315786340494376+001
3	3	.4834442566675688+001	3	4	.1285830362747110+001
4	3	.1511415215861273+001	4	4	.1443059629132175+001
5	3	.3069594460230455+001	5	4	.2317318704508106+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3053186685582263+001	7	4	-.6148470479626041+000
8	3	-.1554060932460036+001	8	4	-.3947065906590448+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 12.7950
ELAPSED TIME WAS 2.4390

THE TERMINAL CONSTRAINT VECTOR

1	.3543537076885874-002	2	.2309774553860570-002
3	-.1264168612138140-002	4	-.4413019196620901-002

THE NORM OF THE TERMINAL CONSTRAINTS .6242161021199833-002

THE A MATRIX

1	1	-.1391361569232885+001	1	2	.5294724581476790+001
2	1	-.1810006433295714+001	2	2	.8260307928105425+000
3	1	.2458130897395349+000	3	2	.4834442566675688+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2390195774391711+001	1	4	-.1369738866755262+000
2	3	.1315786340494376+001	2	4	.1219427048335951+000
3	3	.1285830362747110+001	3	4	.3543537076885874-002
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.6274099140418095+001	1	2	-.7673628086944041+001
2	1	-.7673628086944041+001	2	2	.5210209488959550+002
3	1	-.5391134350221134+001	3	2	.1995858139441027+002
4	1	-.2926583053602419-001	4	2	-.6073795491690776+000
1	3	-.5391134350221134+001	1	4	-.2926583053602419-001
2	3	.1995858139441027+002	2	4	-.6073795491690776+000
3	3	.9100103863953840+001	3	4	-.1623874722258897+000
4	3	-.1623874722258897+000	4	4	.3365335508086855-001

THE CORRECTIONS AT THE 5 TH ITERATION

1	.4382799879546237-002	2	-.1006996231750214-002
3	.3909715075967291-002	4	.1068890832527561-001

BEGINNING THE 6 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 13.0050
ELAPSED TIME WAS .2100

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.3021931707466411-004	6	2	.1000000000000000+001
7	1	-.4948057282884892+000	7	2	.0000000000000000
8	1	-.1078673089805401+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3319482642073252+001

1	1	-.1643661890425153-004	1	2	-.1394502866172372+001
2	1	.8100667134705156+000	2	2	-.1864483536570655+001
3	1	.1523717810411174+001	3	2	.2797642048738974+000
4	1	.2490402217852878+001	4	2	-.1981509606264765+001
5	1	-.7118873438209103+000	5	2	.4169192442773725+001
6	1	-.3021931707466411-004	6	2	.1000000000000000+001
7	1	.7584867497044485+000	7	2	-.3860524782276137+001
8	1	-.6740805432377951+000	8	2	.7973318897854506+001

1	3	.5270784715397122+001	1	4	.2389616139440934+001
2	3	.8289105287211714+000	2	4	.1345030337553331+001
3	3	.4830785318752025+001	3	4	.1272258642729933+001
4	3	.1493350177371745+001	4	4	.1455784106726813+001
5	3	.3003787193476435+001	5	4	.2034101626281530+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3008948286699502+001	7	4	-.5950162772538181+000
8	3	-.1534050720025032+001	8	4	-.3955084250825209+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 15.4430
ELAPSED TIME WAS 2.4380

THE TERMINAL CONSTRAINT VECTOR

1	-.1643661890425153-004	2	-.6056652948429989-004
3	.3881041117436215-004	4	-.3021931707466411-004

THE NORM OF THE TERMINAL CONSTRAINTS .7973657927675546-004

THE A MATRIX

1	1	-.1394502866172372+001	1	2	.5270784715397122+001
2	1	-.1864483536570655+001	2	2	.8289105287211714+000
3	1	.2797642048738974+000	3	2	.4830785318752025+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2389616139440934+001	1	4	-.1393948583534474+000
2	3	.1345030337553331+001	2	4	.1238439172982659+000
3	3	.1272258642729933+001	3	4	-.1643661890425153-004
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.6499205393697581+001	1	2	-.7544133613072361+001
2	1	-.7544133613072361+001	2	2	.5180475322008156+002
3	1	-.5484181043481116+001	3	2	.1985607040463404+002
4	1	-.3652301378107893-001	4	2	-.6321441636249838+000
1	3	-.5484181048481116+001	1	4	-.3652301378107893-001
2	3	.1985607040463404+002	2	4	-.6321441636249838+000
3	3	.9138014352558944+001	3	4	-.1665472890193257+000
4	3	-.1665472890193257+000	4	4	.3476824416303780-001

THE CORRECTIONS AT THE 6 TH ITERATION

1	.3021919026750394-004	2	-.3889340044863507-004
3	.1105287276712826-003	4	.3910861120734608-005

BEGINNING THE 7 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS 15.6530
ELAPSED TIME WAS .2100

THE DEPENDENT VARIABLES AT TIME = .0000000000000000

1	1	.0000000000000000	1	2	.0000000000000000
2	1	.1000000000000000+001	2	2	.0000000000000000
3	1	.1000000000000000+001	3	2	.0000000000000000
4	1	.0000000000000000	4	2	.0000000000000000
5	1	-.1000000000000000+001	5	2	.0000000000000000
6	1	-.1268071601735065-009	6	2	.1000000000000000+001
7	1	-.4948446216889378+000	7	2	.0000000000000000
8	1	-.1078562561077730+001	8	2	.0000000000000000

1	3	.0000000000000000	1	4	.0000000000000000
2	3	.0000000000000000	2	4	.0000000000000000
3	3	.0000000000000000	3	4	.0000000000000000
4	3	.0000000000000000	4	4	.0000000000000000
5	3	.0000000000000000	5	4	.0000000000000000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	.1000000000000000+001	7	4	.0000000000000000
8	3	.0000000000000000	8	4	.1000000000000000+001

THE DEPENDENT VARIABLES AT TIME = .3319486552934372+001

1	1	-.8080414465471319-007	1	2	-.1394668860772473+001
2	1	.8101272560754422+000	2	2	-.1863972791396285+001
3	1	.1523678940437577+001	3	2	.2795225540555539+000
4	1	.2490447207725215+001	4	2	-.1981448756683368+001
5	1	-.7118565780011420+000	5	2	.4169083082093652+001
6	1	-.1268071601735065-009	6	2	.1000000000000000+001
7	1	.7584227926670698+000	7	2	-.3860099736361985+001
8	1	-.6742201594092333+000	8	2	.7973314682733444+001

1	3	.5272713802641023+001	1	4	.2390301516658150+001
2	3	.8285595237660352+000	2	4	.1344725334681354+001
3	3	.4832809048485204+001	3	4	.1272915975232330+001
4	3	.1493458489199005+001	4	4	.1455844626959524+001
5	3	.3005225572023532+001	5	4	.2039010833442484+000
6	3	.0000000000000000	6	4	.0000000000000000
7	3	-.3009758737085179+001	7	4	-.5954870203627850+000
8	3	-.1533775374965061+001	8	4	-.3954961637140963+001

TIME AT COMPLETION OF FORWARD INTEGRATION IS 18.0910
ELAPSED TIME WAS 2.4380

THE TERMINAL CONSTRAINT VECTOR

1	-.8080414465471319-007	2	-.2392455772642215-007
3	-.5956242241783593-007	4	-.1268071601735065-009

THE NORM OF THE TERMINAL CONSTRAINTS .1031958938142468-006

THE A MATRIX

1	1	-.1394668860772473+001	1	2	.5272713802641023+001
2	1	-.1863972791396285+001	2	2	.8285595237660352+000
3	1	.2795225540555539+000	3	2	.4832809048485204+001
4	1	.1000000000000000+001	4	2	.0000000000000000
1	3	.2390301516658150+001	1	4	-.1393236032160158+000
2	3	.1344725334681354+001	2	4	.1238554324718680+000
3	3	.1272915975232330+001	3	4	-.8080414465471319-007
4	3	.0000000000000000	4	4	.0000000000000000

THE A*A MATRIX

1	1	.6497628656500260+001	1	2	-.7547223032165429+001
2	1	-.7547223032165429+001	2	2	.5184406502810879+002
3	1	-.5484401804392589+001	3	2	.1986932062532108+002
4	1	-.3655288780476607-001	4	2	-.6319922780770420+000
1	3	-.5484401804392589+001	1	4	-.3655288780476607-001
2	3	.1986932062532108+002	2	4	-.6319922780770420+000
3	3	.9142142646274473+001	3	4	-.1664739850475708+000
4	3	-.1664739850475708+000	4	4	.3475123456589633-001

THE CORRECTIONS AT THE 7 TH ITERATION

1	.1268071601740951-009	2	.8633038132783541-008
3	.1398768757996405-007	4	-.1454648348999221-007

********** CONVERGENCE HAS BEEN ACHEIVED **********

THE FINAL TIME IS 18.3010

METHOD OF PERTURBATION FUNCTIONS

TWO-POINT BOUNDARY VALUE PROBLEM

INITIAL VALUE OF THE DEPENDENT VARIABLES

1	.0000000000000000	2	.1000000000000000+001
3	.1000000000000000+001	4	.0000000000000000
5	-.1000000000000000+001	6	.0000000000000000
7	-.3958745279999999+000	8	-.8628515199999999+000

DESIRED VALUES OF THE TERMINAL VARIABLES

1	.0000000000000000	2	.8101272799999999+000
3	.1523679000000000+001	4	.0000000000000000

THE INITIAL TIME INTERVAL

FROM .0000000000000000 / TO .3319486499999999+001

THE INTEGRATION STEP SIZE IS .3000000000000000-001

THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS 25

THE ACCURACY REQUIRED FOR TERMINATION IS .1000000000000000-005

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS 2

EVERY 5TH POINT WILL BE PLOTTED FOR EACH 5TH ITERATION

EVERY 0TH POINT WILL BE PRINTED ON EACH ITERATION

PRINT CONTROL SWITCH = 0

PUNCH CONTROL SWITCH = 0

MINIMUM NORM CORRECTION PROCEDURE

VARIABLE ALPHA PROCEDURE

ALPHA = .2000000000000000+001 P = .2000000000000000+001

4 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

GRAVITATIONAL CONSTANT	.1000000000000000+001
INITIAL MASS	.1000000000000000+001
MASS FLOW RATE	.7480039100000000-001
THRUST	.1401296900000000+000

```
                    1 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3319486499999999+001
1       .0000000000000000            2      -.3958745279999999+000
3      -.8628515199999999+000

THE NORM OF THE TERMINAL CONSTRAINTS      .5418663193314779+000

                    2 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3135414365287502+001
1      -.4221201814654489-001        2      -.5010835678174694+000
3      -.9887747258014515+000

THE NORM OF THE TERMINAL CONSTRAINTS      .2292810639409694+000

                    3 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3142364393447604+001
1      -.8807059748364503-002        2      -.5180099183998519+000
3      -.1117882495028377+001

THE NORM OF THE TERMINAL CONSTRAINTS      .1487468624939908+000

                    4 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3154680340397451+001
1      -.4829514768131386-002        2      -.4882545944867250+000
3      -.1080200863589952+001

THE NORM OF THE TERMINAL CONSTRAINTS      .6760633508659172-001

                    5 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3308793733747976+001
1      -.4413019196620901-002        2      -.4937987320567389+000
3      -.1082582804881368+001

THE NORM OF THE TERMINAL CONSTRAINTS      .6242161021199833-002

                    6 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES

                TF =      .3319482642073252+001
1      -.3021931707466411-004        2      -.4948057282884892+000
3      -.1078673089805401+001

THE NORM OF THE TERMINAL CONSTRAINTS      .7973657927675546-004

                    7 TH ITERATION
   DISPLAYING FINAL TIME AND THE LAST  3 DEPENDENT VARIABLES
```

```
                    TF =      .3319486552934372+001
1      -.1268071601735065-009        2      -.4948446216889378+000
3      -.1078562561077730+001

THE NORM OF THE TERMINAL CONSTRAINTS      .1031958938142468-006

***************** CONVERGENCE HAS BEEN ACHEIVED *****************
```

Appendix B

BRACHISTOCHRONE EXAMPLE

1. FORMULATION

The brachistochrone problem is stated as follows: Determine the control history $\theta(t)$ such that a particle falling from x_o, y_o to x_f, y_f under the influence of a constant gravitational force g will do so in a minimum time.

The differential equations of motion are

$$\dot{x} = \sqrt{2g(y-a)} \cos \theta$$

$$\dot{y} = \sqrt{2g(y-a)} \sin \theta$$

where a is a constant $y_o - \frac{v_o^2}{2g}$ and $v = (\dot{x}^2 + \dot{y}^2)^{1/2}$

When the optimization process is applied, the control angle θ is eliminated from the above equations and two additional equations are required to be satisfied (Euler-Lagrange equations). Hence, the nonlinear two-point boundary value problem may be summarized as:

The differential equations for F1 are

$$\dot{z}_1 = \dot{x} = - \frac{\lambda_1 \sqrt{2g(y-a)}}{\sqrt{\lambda_1^2 + \lambda_2^2}}$$

$$\dot{z}_2 = \dot{y} = - \frac{\lambda_2 \sqrt{2g(y-a)}}{\sqrt{\lambda_1^2 + \lambda_2^2}}$$

$$\dot{z}_3 = \dot{\lambda}_1 = 0$$

$$\dot{z}_4 = \dot{\lambda}_2 = \frac{g\sqrt{\lambda_1^2 + \lambda_2^2}}{\sqrt{2g(y-a)}}$$

where $t = t_o$ $\qquad$ $t = t_f$ (unspecified)

$x(t_o) = 0$ $\qquad$ $x(t_f) = 5.0$

$y(t_o) = 1.0$ $\qquad$ $y(t_f) = 8.0$

$$1 - \sqrt{2g(y-a)}\ \sqrt{\lambda_1^2 + \lambda_2^2}\Big|_{t_f} = 0$$

The perturbation equations for F2 are

$$\delta\dot{z}_1 = \delta\dot{x} = -\left[\frac{\lambda_1 g}{\sqrt{\lambda_1^2 + \lambda_2^2}\ \sqrt{2g(y-a)}}\right]\delta z_2 - \left[\frac{\lambda_2^2\sqrt{2g(y-a)}}{(\lambda_1^2 + \lambda_2^2)^{3/2}}\right]\delta z_3$$

$$+ \left[\frac{\lambda_1\lambda_2\sqrt{2g(y-a)}}{(\lambda_1^2 + \lambda_2^2)^{3/2}}\right]\delta z_4$$

$$\delta \dot{z}_2 = \delta \dot{y} = - \left[\frac{\lambda_2 g}{\sqrt{\lambda_1^2 + \lambda_2^2}\ \sqrt{2g(y-a)}} \right] \delta z_2 + \left[\frac{\lambda_1 \lambda_2 \sqrt{2g(y-a)}}{(\lambda_1^2 + \lambda_2^2)^{3/2}} \right] \delta z_3$$

$$- \left[\frac{\lambda_1^2 \sqrt{2g(y-a)}}{(\lambda_1^2 + \lambda_2^2)^{3/2}} \right] \delta z_4$$

$$\delta \dot{z}_3 = \delta \dot{\lambda}_1 = 0$$

$$\delta \dot{z}_4 = \delta \dot{\lambda}_2 = - \left[\frac{g^2 \sqrt{\lambda_1^2 + \lambda_2^2}}{2g(y-a)^{3/2}} \right] \delta z_2 + \left[\frac{\lambda_1 g}{\sqrt{\lambda_1^2 + \lambda_2^2}\ \sqrt{2g(y-a)}} \right] \delta z_3$$

$$+ \left[\frac{\lambda_2 g}{\sqrt{\lambda_1^2 + \lambda_2^2}\ \sqrt{2g(y-a)}} \right] \delta z_4$$

The terminal constraints for F3 are

$$h_1 = x(t_f) - 5.0$$

$$h_2 = y(t_f) - 8.0$$

$$h_3 = 1 - \sqrt{\lambda_1^2 + \lambda_2^2}\ \sqrt{2g(y-a)}\Big|_{t_f}$$

The partial derivative of the terminal constraints with respect to the dependent variables and the time rates of change of the terminal constraints for F4 are

$$\left[\frac{\partial h}{\partial z}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ D_1 & D_2 & D_3 & D_4 \end{bmatrix}_{t_f}$$

where

$$D_1 = 0 = -\dot{z}_3(t_f)$$

$$D_2 = -g\,\frac{\sqrt{\lambda_1^2 + \lambda_2^2}}{\sqrt{2g(y-a)}} = -\dot{z}_4(t_f)$$

$$D_3 = -\frac{\lambda_1\sqrt{2g(y-a)}}{\sqrt{\lambda_1^2 + \lambda_2^2}} = \dot{z}_1(t_f)$$

$$D_4 = -\frac{\lambda_2\sqrt{2g(y-a)}}{\sqrt{\lambda_1^2 + \lambda_2^2}} = \dot{z}_2(t_f)$$

and

$$\dot{h} = \begin{bmatrix} \dot{x}(t_f) \\ \dot{y}(t_f) \\ 0 \end{bmatrix}$$

2. MODIFICATIONS FOR CUR SYSTEM

```
$JOB  LEC1SW S WILLIAMS L09856 ED02  E131  7006  C   3   2        4020    4 B
'N HDG       INPUT 10615.
' ASG P=10615
' XQT CUR
  TRW P
  IN P
  TRI P
  TOC
'T FOR,* MAIN,MAIN,MAIN/B
-1,1
       PARAMETER N=4,NQ=3,KON=2,NS3=2
'T FOR,* ABAM,ABAM,ABAM/B
-2,2
       PARAMETER N=4,NQ=3
'T FOR,* CONVRG,CONVRG,CONVRG/B
-2,2
       PARAMETER N=4,NQ=3
'T FOR,* CRDPCH,CRDPCH,CRDPCH/B
-2,2
       PARAMETER N=4
'T FOR,* FPLEW,FPLEW,FPLEW/B
-2,4
       PARAMETER N=4,NQ=3,NS3=2, KON=2
       DOUBLE PRECISION DEG
       DATA DEG/57.29577951308232D0/
-61,62
                 X(1)=DEP(1,1)
                Y(2)=-DEP(2,1)
'T FOR,* FPLOT,FPLOT,FPLOT/B
-2,2
       PARAMETER NS3=2
'T FOR,* FPRNT,FPRNT,FPRNT/B
-2,2
      PARAMETER N=4, NQ=3
'T FOR,* F1,F1,F1/B
-2,2
       PARAMETER N=4,KON=2
-5,6
       DOUBLE PRECISION WOED,WOGD,GM,AI,DSQRT
       EQUIVALENCE (CON(1),GM),(CON(2),AI)
-9,19
       WOED=DSQRT(2.D0*GM*(TY(2)-AI))
      WOGD=DSQRT(TY(3)**2+TY(4)**2)
      P(1)=-TY(3)*WOED/WOGD
      P(2)=-TY(4)*WOED/WOGD
      P(3)=0.D0
      P(4)= GM*WOGD/WOED
'T FOR,* F2,F2,F2/B
-2,2
       PARAMETER N=4,KON=2
-5,7
       DOUBLE PRECISION WOED,WOGD,GM,AI,DSQRT
       EQUIVALENCE (CON(1),GM),(CON(2),AI)
       DOUBLE PRECISION R,R2,R3,R4
-27,59
       WOED=DSQRT(2.D0*GM*(S(2,J)-AI))
       WOGD=SQRT(S(3,J)**2+S(4,J)**2)
      R=WOED**3
      R2=WOGD**3
       R3=S(3,J)**2
       R4=S(4,J)**2
       A(1,2)= -S(3,J)*GM/(WOED*WOGD)
```

```
      A(1,3)=-R4*WOED/R2
      A(1,4)= S(4,J)*S(3,J)*WOED/R2
      A(2,2)= -S(4,J)*GM/(WOED*WOGD)
      A(2,3)= S(3,J)*S(4,J)*WOED/R2
      A(2,4)= -R3*WOED/R2
      A(4,2)= -GM*GM*WOGD/R
      A(4,3)= S(3,J)*GM/(WOGD*WOED)
      A(4,4)= S(4,J)*GM/(WOGD*WOED)
'T FOR,* F3,F3,F3/B
-2,2
      PARAMETER N=4,NQ=3, KON=2
-15,18
     H(1)=DEP(1,1)-XF(1)
     H(2)=DEP(2,1)-XF(2)
     H(3)=0.D0
     DO 20 I=1,NQ1
     H(3)=H(3)+YPR(I,I2B,1)*DEP(NQ1+I,1)
  20 CONTINUE
     H(3)=1.D0 +H(3)
'T FOR,* F4,F4,F4/B
-2,2
      PARAMETER N=4,NQ=3, KON=2
-23,26
      B(1,1)=1.D0
      B(2,2)=1.D0
     B(3,2)=-YPR(4,I2B,1)
     B(3,3)= YPR(1,I2B,1)
     B(3,4)= YPR(2,I2B,1)
-30,33
     A(1,NQ)=YPR(1,I2B,1)
     A(2,NQ)=YPR(2,I2B,1)
     A(3,NQ)=0.D0
'T FOR,* F5,F5,F5/B
-2,2
      PARAMETER NS3=2
'T FOR,* F6,F6,F6/B
-2,2
      PARAMETER N=4,NQ=3,KON=2
'T FOR,* F7,F7,F7/B
-2,2
      PARAMETER N=4,NQ=3
'T FOR,* INTEG,INTEG,INTEG/B
-2,2
     PARAMETER N=4
'T FOR,* INTRK5,INTRK5,INTRK5/B
-2,2
      PARAMETER N=4,NQ=3
'T FOR,* ITER,ITER,ITER/B
-2,2
     PARAMETER N=4, NQ=3
'T FOR,* MLTPLY,MLTPLY,MLTPLY/B
-2,2
      PARAMETER NQ=3
' XQT CUR
  TOC
' XQT MAIN/B
```

```
                    X-DISPLACEMENT
                    Y-DISPLACEMENT
        0.0                                 1.0
   -.0357353733314653          -.1726379971977703
        5.0                                 8.0
        0.0
        0.0                     .6076633560035577
        0.006                   0.0000000001
   2   50    1    1  200       1    1    0    0
GRAVITATIONAL CONSTANT                      32.1741
INITIAL CONSTANT                            0.5
' EOF
```

METHOD OF PERTURBATION FUNCTIONS

TWO-POINT BOUNDARY VALUE PROBLEM

INITIAL VALUE OF THE DEPENDENT VARIABLES

1	.0000000000000000	2	.1000000000000000+001
3	-.3573537333146529-001	4	-.1726379971977703+000

DESIRED VALUES OF THE TERMINAL VARIABLES

1	.5000000000000000+001	2	.8000000000000000+001
3	.0000000000000000		

THE INITIAL TIME INTERVAL

FROM .0000000000000000 / TO .6076633560035576+000

THE INTEGRATION STEP SIZE IS .5999999999999999-002

THE MAXIMUM NUMBER OF ITERATIONS ALLOWED IS 50

THE ACCURACY REQUIRED FOR TERMINATION IS .1000000000000000-009

THE NUMBER OF ITERATIONS WITH THE ADAMS-MOULTON CORRECTOR IS 2

EVERY 1TH POINT WILL BE PLOTTED FOR EACH 1TH ITERATION

EVERY 200TH POINT WILL BE PRINTED ON EACH ITERATION

PRINT CONTROL SWITCH = 1

PUNCH CONTROL SWITCH = 0

NORMAL CORRECTION PROCEDURE

2 SPECIAL INPUT CONSTANTS AND THEIR IDENTIFICATIONS

GRAVITATIONAL CONSTANT	.3217410000000000+002
INITIAL CONSTANT	.5000000000000000+000

```
                    BEGINNING THE   1 TH ITERATION
TIME AT THE COMMENCEMENT OF FORWARD INTEGRATION IS        .0810
                    ELAPSED TIME WAS        .0810

  THE DEPENDENT VARIABLES AT TIME =     .0000000000000000


1  1     .0000000000000000          1  2     .0000000000000000
2  1     .1000000000000000+001      2  2     .0000000000000000
3  1    -.3573537333146529-001      3  2     .1000000000000000+001
4  1    -.1726379971977703+000      4  2     .0000000000000000


                    1  3      .0000000000000000
                    2  3      .0000000000000000
                    3  3      .0000000000000000
                    4  3      .1000000000000000+001

  THE DEPENDENT VARIABLES AT TIME =     .6076633560035576+000


1  1     .4999999999999774+001      1  2    -.9064133566788512+002
2  1     .6000000000000303+001      2  2     .1148759769856110+003
3  1    -.3573537333146529-001      3  2     .1000000000000000+001
4  1    -.2819650908506611-001      4  2    -.6200888295657205+000


                    1  3      .1877417351888520+002
                    2  3     -.2379487288617123+002
                    3  3      .0000000000000000
                    4  3      .2918527621568634+000

   TIME AT COMPLETION OF FORWARD INTEGRATION IS        .9530
                    ELAPSED TIME WAS        .8720

                    THE TERMINAL CONSTRAINT VECTOR

 1     -.2251185349244622-012       2      .3036321194471724-012
 3     -.6320638457069094-013

 THE NORM OF THE TERMINAL CONSTRAINTS     .3832308257134548-012

            THE CORRECTIONS AT THE  1 TH ITERATION

 1     -.2170286656173745-015       2      .1141735878249916-013
 3     -.5163256476650848-015

***************** CONVERGENCE HAS BEEN ACHEIVED *****************

                    THE FINAL TIME IS       1.1900
```

NOTES

NOTES (Concluded)