This report is a description of, and user's instructions for a program for the solution of linear and nonlinear first-order ordinary differential equations. The program has a new integration algorithm for the solution of initial value problems which is particularly efficient for the solution of differential equations with a wide range of eigenvalues. The program in its present form will handle up to ten state variables, but is being expanded to handle up to fifty state variables.

Prepared by:
D. G. Fitzjerrell

CONCURRENCES

DISTRIBUTION
GE/AGS: Central Product File
R. C. Croston, Ph.D.
V. J. Marks
R. F. Hassell

NASA/JSC: Technical Library/JM6

Page No.
1 of 1
NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
A. IDENTIFICATION

Program Name: Ordinary Differential Equation Solver - HMS

Author: H. Sloate, G.E. Co., Electronics Laboratory Syracuse, N. Y.


Programmer Contact: V. J. Marks, GE/MSC, Houston

Date of Issue: May 1, 1973

B. GENERAL DESCRIPTION

The program is designed to provide numerical solution to systems of linear and nonlinear first-order ordinary differential equations. The program contains a new integration algorithm for the solution of initial value problems and is particularly efficient for solving differential equations having a wide range of eigenvalues. For the classical methods, the integration step size is limited more by stability considerations than by accuracy. The new implicit fourth-order linear multistep method based on Gear's formula utilized in this program does not become numerically unstable as the time step size becomes large. Since the integration formula is a multistep method, it must be started by some other means. Gill's fourth-order Runge Kutta method is used to calculate the three points in addition to the initial conditions needed to start the multistep process. After the starting procedure is carried out and an acceptable time step size found, the step size is still kept small enough for accuracy. If it is too large, it is halved. If it is too small, it can cause excessive running times and an appropriate test is performed to detect this event and the step size is doubled.

C. USAGE AND RESTRICTIONS

Machine and Compiler Required: UNIVAC 1108 and FORTRAN

Peripheral Equipment Required: Card Reader; Line Printer

Approximate Amount of Memory: 6,027 Required
D. PARTICULAR DESCRIPTION

Consider the system of first-order differential equations given in (1):
\[ \dot{y} = f(y, t), \quad y(0) = y_0 \]  (1)

The function \( f \) may be a non-linear, time-varying function of \( y \) and hence it may be difficult or impossible to find an analytic solution to eq. (1) if a solution exists. If one does exist, the above system of equations may be integrated point by point in time to obtain an approximation to the solution of eq. (1), \( y(t) \).

There are many formulas used for the integration of ordinary differential equations. Some are designed to integrate differential equations of high order while others are used for a system of first-order differential equations such as (1).

The form given in eq. (1) is general because differential equations of any order can be rewritten as systems of first-order equations. Thus it is of interest to examine integration formulas which can be used to solve eq. (1).

Classical integration techniques include the linear multi-step methods, the predictor-corrector methods, and the Runge Kutta methods. One might ask why new methods are being invented when we have such a repertoire of existing methods at our disposal. The following discussion will point out the shortcoming common to all classical methods.

1. STIFF DIFFERENTIAL EQUATIONS

A stiff set of linear ordinary differential equations is defined as one which has very large and very small eigenvalues. The term stiff probably comes from structural engineering where stiff members gave rise to large eigenvalues in the differential equation formulation.

When numerically integrating stiff differential equations, one encounters two conflicting requirements: a) the time step size must be large to reduce the number of steps taken and consequently the labor required to obtain the solution, and b) the time step must be small to prevent the integration algorithm from becoming numerically unstable. Stiff differential equations aggravate the above conflict because the maximum allowable time step for numerical stability is inversely proportional to the largest eigenvalue. Many time steps must be taken to display the solution associated with the small eigenvalues of the system. The combination of small step size and long running time causes the solution to be calculated at great many points.

To see how a numerical integration formula can become unstable let us use Euler's forward integration formula given in eq. (2) to solve the linear first-order differential equation in eq. (3).

\[
\begin{align*}
  y_{n+1} &= y_n + h \dot{y}_n \\
  \dot{y} &= -\lambda y, \quad y(0) = y_0
\end{align*}
\]  (2)  (3)
Substituting eq. (3) into (2) and solving we obtain the recursion relationship:

\[ y_{n+1} = (1 - \lambda h)y_n \]  

(4)

The solution to eq. (4) is

\[ y_n = (1 - \lambda h)^n y_0 \]  

(5)

For the sequence \([Y_n]\) to be bounded it follows that:

\[ \begin{align*}
1 - \lambda h &\leq 1 \\
0 &\leq \lambda h \leq 2 \\
0 &\leq h \leq \frac{2}{\lambda}
\end{align*} \]  

(6)

That is, if the numerical solution is to have the same property as the actual solution to eq. (3) (i.e., boundedness) then there is an upper limit on \(h\) which is inversely proportional to the largest eigenvalue in the system of equations being integrated. This is generally true of all classical integration formulas.

2. **Implicit Integration Formulas**

Much research has been done in recent years to circumvent the numerical instability problem. One approach is to examine the equations being integrated and to modify them to delete the troublesome eigenvalues. However, this may be time consuming and not completely general. Another approach is to use a non-linear integration formula which uses exponentials rather than polynomials to fit the functions being integrated. This type of formula has excellent stability properties and is exact for linear systems (no truncation error). However, the Jacobian of the system of differential equations must be calculated and the exponential matrix generated. Here we have solved the numerical instability problem but have replaced it with a great deal of computation per time step. A tradeoff between stability and computation per time step can be made by using the convergence properties of the exponential power series as is done in the CIRCUS network analysis program.

Lacking a complete solution to the problem, let us turn our attention to linear multi-step integration formulas of the form:

\[ y_{n+1} = \sum_{i=0}^{k-1} a_i y_{n-i} + h \sum_{i=-1}^{k-1} \beta_i y_{n-i} \]  

(7)

The \(a\)'s and \(\beta\)'s may be determined by requiring that the formula be exact up to a certain order, \(p\), if the function, \(y(t)\), being integrated is a polynomial in time of order \(p\) or less. If there are more \(a\)'s and \(\beta\)'s than necessary to make the formula exact up to and including order \(p\), then the remaining arbitrary \(a\)'s and \(\beta\)'s may be chosen to optimize other properties of the integration formula.

If \(\beta_{-1} \neq 0\) the integration formula is said to be an implicit formula because the unknown, \(y_{n+1}\), appears on both sides of the equation. These equations have the same form as the corrector formulas in the predictor-corrector method. If \(\beta_{-1} = 0\) the integration formula is an explicit formula, similar to the predictor formulas in the predictor-corrector method.
The implicit equations can have numerical stability for the time step size, \( h \), arbitrarily large, as the following example shows. Consider Euler's backward formula which is an implicit formula of order one and step number \( k = 1 \).

\[
y_{n+1} = y_n + hy_{n+1}
\]

If we use eq. (8) to solve eq. (3) we obtain the difference equation:

\[
y_{n+1} = (1 + \lambda h)^{-1} y_n
\]

The solution to eq. (9) is

\[
y_n = (1 + \lambda n)^{-n} y_0
\]

If \( \text{Re}[\lambda] \geq 0 \), then \( |y_n| \) is a bounded sequence for all \( h \geq 0 \). Thus there is no upper limit on time step size due to numerical instability problems. Here is one formula, at least, which has the desired numerical stability properties.

The question arises as to whether there are more formulas with the above stability properties. Before answering this question let us present the concept of A-stability. A \( k \) step method is called A-stable if all solutions of eq. (7) tend to zero, as \( n \to \infty \), when the method is applied with fixed positive \( h \) to eq. (3) where \( \lambda \) is a complex constant with positive real part. A-stability requires that if the solution to the differential equation is stable then the corresponding solution to the difference equation be stable also. The example given in eq. (8) is an A-stable method. Unfortunately there are not too many more of them.

Dahlquist proved two theorems important to our discussion:

**Th. 1.** An explicit linear multistep method cannot be A-stable.

**Th. 2.** The order, \( p \), of an A-stable linear multistep method cannot exceed 2.

Theorem 2 indicates that if we require A-stability we will be able to use only formulas of low order and hence limited accuracy. If the restriction of A-stability is removed, we can find high-order formulas with acceptable numerical stability properties.

In Appendix I it is shown that if we want an integration formula which remains stable as \( h \to \infty \), we must use an implicit formula and its maximum order cannot exceed its step number \( k \). Such an integration formula will not be A-stable, however, if \( p > 2 \).

### 3. INTEGRATION FORMULA SELECTION

As \( h \to \infty \) it would be good to have the roots of the integration formula tend to zero so that the unimportant large eigenvalue modes of a system of differential equations would rapidly die out as the time step size is increased. This is the approach taken by Gear in obtaining implicit integration formulas of order 2 through 6. In eq. (7) there are \( 2k+1 \) arbitrary constants. If a \( k \)th order fit is required, \( k+1 \) of the constants are fixed. If the \( k \) roots of \( \sum_{i=1}^{k} \beta_i z^i = 0 \) are all to be zero, the remaining \( k \) constants of eq. (7) are fixed and the integration formula is unique. The formulas of order 2 through 6 are given in Appendix II. Generally speaking it is desirable to use as high an order formula as possible to achieve greater accuracy. The higher order formulas have the disadvantage in that more past values must be stored and more manipulation is required when
the step size is halved or doubled. A fourth-order formula was chosen for integrating the systems of differential equations as a compromise between accuracy and complexity. It is different from Gear's fourth-order formula in that its coefficients were selected to achieve a compromise between truncation error and roots close to zero as $h \to \infty$. This was done by minimizing a performance index using the Fletcher Powell method. The performance index was a weighted sum of the truncation error given in eq. (12) and the square of the sum of the squares of the roots of eq. (7) as $h \to \infty$.

4. TRUNCATION ERROR

The truncation error of the integration formula in eq. (7) is due to the fact that the formula has only a finite number of terms. This error would be present even if there were no roundoff errors in the computer due to using numbers with a finite number of bits. An expression for the truncation error can be obtained by integration by parts. (See Appendix III.) It is given by

$$T_n = \frac{y^{(k+1)}(N)}{k!} \int_{(n-k+1)h}^{(n+1)h} G(s) ds$$

(11)

where $G(s)$ is called the influence function and is of the same sign over the interval from $(n+1-k)h$ to $(n+1)h$. The $k+1$th derivative is evaluated at $N$ which is somewhere in the interval from $(n-k+1)h$ to $(n+1)h$. If $G(s)$ changes sign over the interval, the First Mean Value Theorem which was used in obtaining eq. (11) does not apply and an estimate of the error can be obtained from:

$$T_n = \left| \frac{y^{(k+1)}(N)}{k!} \right| \int_{(n-k+1)h}^{(n+1)h} |G(s)| ds$$

(12)

Equations (11) and (12) were used to obtain the truncation error of the integration formulas studied.

If $G(s)$ is of the same sign over the interval and eq. (11) applies, an alternate shorter method can be used to compute $T_n$. Assume the error is given by

$$T_n = \frac{E_k}{(k+1)!} h^{k+1} y^{(k+1)}(N)$$

(11a)

Assume $Y(t) = t^{k+1}$ and substitute into eq. (7). Since eq. (11a) is the error in eq. (7) $E_k$ can be evaluated. This is the standard way to find error terms in quadrature formulas. Note, however, that it is valid only if $G(s)$ is of the same sign over the interval covered by the integration formula. (See Appendix III for an example of the procedure for calculating $E_k$.)

5. STABILITY BOUNDARIES

Let $\dot{y} = qy$. The roots of the difference equation given in eq. (7) are the zeros of

$$\lambda^{k-1} \left( \alpha - qhj\right) + 0, \quad \alpha = -1$$

(13)
Solving for \( qh \) gives

\[
qh = \frac{\sum_{i=-1}^{k-1} \beta_j \zeta^{k-1-i}}{k-1} \quad (14)
\]

If there is some value of \( qh \) which causes \( \zeta \) to have unit magnitude, then we can let \( \zeta = e^{a \theta} \) and let \( \theta \) vary from 0 to \( \pi \). This will map the corresponding stability boundary in the \( qh \) plane. If the stability boundary lies wholly in the plane \( \text{Re} \{ qh \} \geq 0 \), then the integration formula being considered will be A-stable. If not, then there are some values of \( qh \) in the plane \( \text{Re} \{ qh \} < 0 \) for which the roots are not less than one in magnitude and by definition the method is not A-stable.

6. COMPARISON OF TWO FOURTH-ORDER FORMULAS

Gear's fourth-order formula has a truncation error given by:

\[
|T_n| = \frac{2.304}{4!} h^5 |y^{(5)}(\eta)| \quad (15)
\]

The truncation error of the optimized fourth-order formula given in Appendix II is:

\[
|T_n| = \frac{1.932}{4!} h^5 |y^{(5)}(\eta)| \quad (16)
\]

These expressions can be obtained by integrating the influence functions shown in Figure 1. See Appendix III.

The stability boundaries in the \( qh \) plane are shown in Figures 2 and 3. Both methods are A(\( \alpha \)) stable in the sense of Widlund. Gear's method has an \( \alpha = 72 \) degrees. The optimized method has an \( \alpha = 63 \) degrees. It can be seen that there is a trade-off between truncation error and the size of the stability sector in the \( qh \) plane. The optimized formula is more accurate than Gear's method but its stability sector is smaller.

7. SOLUTION OF THE IMPLICIT EQUATION

The integration formula given in eq. (7) is implicit if \( \beta_{-1} \neq 0 \). Since the unknown, \( y_{n+1}^{(k)} \), appears on both sides of the equal sign, the equation will have to be solved iteratively. One approach which is used in the predictor-corrector methods is the method of successive substitution (Picard's Method). The \( k \)th estimate of \( y_{n+1}^{(k+1)} \) is used on the right side of eq. (7) to evaluate \( y_{n+1}^{(k+1)} \), and this is used to evaluate the \( k+1 \)th estimate of \( y_{n+1}^{(k+1)} \).

\[
y_{n+1}^{(k+1)} = \beta_1 y_{n+1}^{(k)} + \sum_{i=-1}^{k-1} \alpha_i y_{n+1}^{(k)} + h \beta_1 y_{n+1}^{(k)} \quad (17)
\]
Figure 1. Influence Function for Two Fourth Order Formulas

Figure 2. Stability Region for Gear's Formula
To see if this can be of use let us examine the linear case $\dot{y} = qy$. Letting

$$K = \sum_{i=0}^{k-1} \alpha_i y_{n-i} + h\beta_i y_{n-1}$$

because $K$ does not change during the iteration, we find that eq. (17) is of the form:

$$y_{n+1}^{(k+1)} = h\beta_{-1}qy_{n+1}^{(k)} + K$$

(18)
If \( y^{(o)}_{n+1} \) is the initial guess for \( y_{n+1} \), the solution to eq. (18) is:

\[
y^{(k)}_{n+1} = (qh)^k y^{(o)}_{n+1} + \frac{1 - (qh)^k}{1 - qh} K_n
\]

This is a divergent sequence if \(|qh| > 1\). This is the same restriction we were trying to avoid by using implicit methods in the first place. It can be shown that this restriction holds for systems of equations and nonlinear equations as well.

The Newton-Raphson method can be used to solve implicit equations such as eq. (7). It does not limit the size of \( qh \). If eq. (7) is applied to systems of equations such as:

\[
(Y_{n+1}, \ldots, Y_{n+1}) = \sum_{i=0}^{k-1} (\alpha_i Y_{n-i} + h \beta_i Y_{n-i})
\]

it can be seen that \( Q (Y_{n+1}) \) will be zero at the solution. If \( Q \) is expanded in a Taylor series and only the first term is kept, we have

\[
Q (y^{(k)}_{n+1} + \Delta y_{n+1}) = Q (y^{(k)}_{n+1}) + J \cdot \Delta y_{n+1}
\]

Solving eq. (21) for \( \Delta y_{n+1} \) gives

\[
\Delta y_{n+1} = -J^{-1} Q (y^{(k)}_{n+1})
\]

where \( J \) is the Jacobian of \( Q \). The Jacobian can be found from eq. (20) and is

\[
J = J - \beta_1 h A
\]

where \( A \) is given by

\[
\begin{bmatrix}
\frac{\partial f_1}{\partial y_1} \\
\frac{\partial f_1}{\partial y_2} \\
\vdots \\
\frac{\partial f_n}{\partial y_n}
\end{bmatrix}, \quad y_1 = f_1 \\
y_2 = f_2 \\
\vdots \\
y_n = f_n
\]

If the differential equations being integrated are linear, the matrix \( A \) does not change during the problem and needs to be calculated only once. For very nonlinear equations \( A \) may change rapidly and a new Jacobian may have to be calculated at each time step. In practice, however, a new Jacobian is not calculated this often. If the number of iterations required for the Newton method to converge exceeds a certain number, the Jacobian is recalculated using difference techniques, it is inverted, and a new \( \Delta y_{n+1} \) is found. If the number of iterations per step does not exceed the limit, the inverse Jacobian from the previous step is used. In this manner considerable time is saved which would be used in evaluating the derivatives \( \frac{\partial f_k}{\partial y_i} \).
If after calculating a new Jacobian the Newton method still fails to converge, the step size $h$ is halved and eq. (7) is applied once more. If after $h$ has been halved a certain number of times there is still no convergence, the simulation is terminated and an error message printed out.

If the system of differential equations being integrated is very stiff and the transient has died out, $qh >> \lambda$ for all the eigenvalues of the system.

$$[1 - \beta^{-1} h A]^{-1} \approx -\frac{1}{\beta^{-1} h} A^{-1}. \tag{24}$$

Equation (24) shows that inverting the Jacobian is almost the same as inverting $A$ which has extreme ranges in eigenvalues. Due to roundoff in the computer, $A$ seems nearly singular and gives very inaccurate results when it is inverted. An iterative scheme to correct the elements of the inverse matrix has been used to overcome the problem. It greatly reduces the number of iterations used for convergence of Newton's method in some cases.

8. SIMULATION PROGRAM ORGANIZATION AND LOGIC

a. Starting Procedure

Since the integration formula in eq. (7) is a multistep method it must be started by some other means. Gill's fourth-order Runge Kutta method was chosen to calculate the three points in addition to the initial conditions needed to start the multistep method. Two more points are calculated using the multistep method and these six points are used to approximate the fifth derivative of $y(t)$ by difference techniques. From this the truncation error is calculated and the accuracy test is made to determine if the time step size is too large. If it is, the step size is reduced by a factor of ten and the starting procedure is re-initialized. The error test is:

$$|T_n| < (1 + |y_{n+1}|) \times \text{ELIM} \tag{25}$$

If this inequality is satisfied, the time step size is acceptable. ELIM is data input by the user. If $y(t)$ is an extremely large number, roundoff error in the computer will keep $T_n$ from being less than ELIM. The $|y_{n+1}|$ term is included to prevent the occurrence of this situation.

b. Time Step Size Control

After the starting procedure has been successfully carried out and an acceptable time step size, $h$, found, eq. (25) is still used to keep $h$ small enough for accuracy. If it is too large, it is halved. If it is too small, it can cause excessive running times. Hence the test in eq. (26) is made.

$$|T_n| < (1 + |y_{n+1}|) \times \text{ELIM/40}. \tag{26}$$

If this inequality is satisfied, the step size is doubled. The factor of 40 was chosen because the error is proportional to $h^9$. Doubling $h$ multiplies the truncation error by 32 if $y^5(\eta)$ remains constant. The factor of 40 was chosen so that eq. (25) would still be satisfied after $h$ was doubled.
Since six points are necessary to calculate the fifth derivative, it is necessary to have five accurate past points before $y_{n+1}$ is calculated. Before doubling $h$, eight past values of $y(t)$ are required so that there will be five past values of $y(t)$ when $y_{n+1}$ is calculated on the next step. (See Figure 4.) The step-size doubling is inhibited until eight past points have been calculated.

If $y_{n+1}$ is calculated and the truncation error is too large, the step size is halved. (See Figure 4.) $y_{n-1}$ and $y_{n-3}$ are found by interpolation. $y_{n-1}$ and $y_{n-3}$ are calculated from these values using the state equations. The old $y_{n-1}$, $y_{n-3}$ are placed in $y_{n-2}$, $y_{n-2}$. The old $y_{n-2}$, $y_{n-2}$ are placed in $y_{n-4}$, $y_{n-4}$. A new $y_{n+1}$ is calculated from the integration formula. If the truncation error is still too large, the step size is halved again and the process repeated.

$$n-4 \quad n-3 \quad n-2 \quad n-1 \quad n \quad n+1$$

$$\begin{array}{cccccc}
\times & \times & \times & \times & \times & \times \\
\end{array}$$

$\text{next step}$

$$n-7 \quad n-6 \quad n-5 \quad n-4 \quad n-3 \quad n-2 \quad n-1 \quad n \quad n+1$$

DOUBLING THE STEP SIZE

$$n-4 \quad n-3 \quad n-2 \quad n-1 \quad n \quad n+1$$

$$\begin{array}{cccccc}
\times & \times & \times & \times & \times & \times \\
\end{array}$$

$\text{inaccurate point}$

$$\text{halve step size}$$

$$n-4 \quad n-3 \quad n-2 \quad n-1 \quad n \quad n+1$$

HALVING THE STEP SIZE

Figure 4. Logic for Changing Step Size

c. The Newton Iteration

The Newton Raphson method is used to drive $Q$ to zero. An initial estimate of the solution, $y_{n+1}^{(0)}$, is calculated by extrapolation using the five previous points which have been stored. Using this estimate $Q^{(0)}$ is calculated and $\Delta y_{n+1}$ is calculated.

$$y_{n+1}^{(k+1)} = y_{n+1}^{(k)} + T \cdot \Delta y_{n+1}$$

(27)
The new value of $y_{n+1}^{(1)}$ is computed using eq.(27) with $T_1 = 1$. $Q^{(1)}$ is evaluated and if $Q^{(1)\top} Q^{(1)} < Q^{(0)\top} Q^{(0)}$, a new $\Delta y_{n+1}$ is calculated and the process is repeated. If not, $T_1$ is decreased by a factor of 10 and eq. (27) is applied again with the same $\Delta y_{n+1}$. This process is repeated up to NDEX times. If there is still no success, a new Jacobian is calculated and a new $\Delta y_{n+1}$ is used in eq. (27). If there is no success even then, $h$ is halved and the whole procedure is tried again. $h$ can be halved up to NHLIM times. If no success is obtained, an error message is printed and the program terminated.

New $\Delta y_{n+1}$ values are checked in the inequality

$$|\Delta y| < (SC + |y|) \times XNLIM$$

If eq. (28) is satisfied for all the state variables, the iteration has been completed successfully and the last estimate of $y_{n+1}$ is taken to be the solution of the implicit integration formula. SC is usually set to 1. and XNLIM is usually set to 0.1 * ELIM.

Each time a $\Delta y_{n+1}$ step has been successfully carried out, the estimate of the inverse Jacobian is updated using Broyden's scheme. This updating keeps the estimate of the inverse Jacobian current and lessens the need for calculating new Jacobians.

F. DESCRIPTION OF INPUT

The differential equation solver is a stand-alone program. Flow charts of the program are shown in Appendix IV. It needs two sets of input from the user. The first set is input data which is read in on three cards.

The first card has a format (2F8.1, 15) and reads the variables TF, PT, NV. TF is the final time for the problem. When the solution time reaches this value, execution will terminate. PT is the number of points in time the user wants printed out. For example, if TF = 10. and PT = 5., the state variables would be printed out at t = 2., 4., 6., 8., and 10. seconds. NV is the number of state variables.

The second data card has a format (10F8.2). The initial values of the state variables (up to 10) are input here. If there are less than 10 state variables, leave the unused columns blank.

The third card has a format (4E7.1) and reads the variables YLIM, ELIM, XNLIM, SC. YLIM is usually set at about $10^{-5}$ and governs how the state variables are perturbed to calculate the elements in the Jacobian matrix. The program uses YLIM as follows:

$$\begin{align*}
\text{If } |y_n| < \text{YLIM, perturb } y_n \text{ by } 10^{-4}. \\
\text{If } |y_n| < \text{YLIM, perturb } y_n \text{ by } 10^{-3} \times y_n
\end{align*}$$

ELIM controls the truncation error of the integration formula. See eqs. (25) and (26). It is usually set at about $10^{-3}$. If a more accurate solution is required, it can be set smaller. However, a smaller ELIM requires more time steps to complete a simulation. XNLIM controls the convergence requirements for the Newton iteration. See eq.(28). It is usually set to 0.1 * ELIM.
The other input required of the user is a subroutine in FORTRAN which describes the system of equations being integrated. An example is shown below. The maximum number of state variables is 10. The state variables are contained in the array X. The first derivatives of X are stored in the array D. The variable NUM is used to count how many times the state equations have been evaluated during a simulation.

```fortran
CSTATE
SUBROUTINE STATE(X,D)
COMMON YO,DY,DH,T,H,NV,NUM
DOUBLE PRECISION YO(10),DY(10),DH(10)
DOUBLE PRECISION X(1),D(1)
1 D(2)=X(1)-X(2)
2 D(1)=-10000.*(X(1)+D(2))
3 NUM = NUM+1
4 RETURN
END
```

F. DESCRIPTION OF OUTPUT

Since the integration formula is a variable step method, the variables may not be calculated at the values of time at which printout is desired. Hence, it is necessary to use Lagrangian interpolation to find the values of the state variables at the desired time. See Appendix B for Sample Output.

G. INTERNAL CHECKS AND EXITS

None.

H. INDEPENDENT SUBROUTINES

None

I. SYSTEM SUBROUTINES

No special subroutine.

J. COMPLETION OR FINAL CHECKOUT DATE

April 25, 1973

Source listing and sample cases are given in Appendix B.
BIBLIOGRAPHY

APPENDIX A

MATHEMATICAL PROOFS

GEAR'S FORMULA

TRUNCATION ERROR
MATHEMATICAL PROOFS

DEFINITION - A linear multistep method is said to be consistent if it is of at least order one.

If a linear multistep method is consistent, $\sum_{i=0}^{k} \alpha_i \xi^i$ has a simple root at $\xi = 1$. Let $y = K$, a constant. $\therefore y = 0$. Substituting this into

$$\sum_{i=0}^{k} \alpha_i y_{n+i} - h \sum_{i=0}^{k} \beta_i y_{n+i} = 0$$

results in

$$\sum_{i=0}^{k} \alpha_i = 0$$

(2)

If there are two roots at $\xi = 1$, the solution of the difference equation is

$$y_n = K_1 + K_2 n + K_3 \xi + \ldots + K_k \xi^k$$

(3)

where the $K_i$'s are determined by initial conditions. The sequence $\{y_n\}$ is not convergent as $n \to \infty$ because of the term containing $n$. Hence the method will not be convergent if $\sum_{i=0}^{k} \alpha_i \xi^i$ has more than one zero at $\xi = 1$. (See Henrici, P. Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962, pp. 217-218.)

We will now show that consistency requires that $\sum_{i=0}^{k} \beta_i \neq 0$. Let $y = 1$, $y(0) = 0$. The solution to the differential equation is $y(t) = t$. Equation (1) becomes

$$\sum_{i=0}^{k} \alpha_i \cdot 1 - \sum_{i=0}^{k} \beta_i = 0$$

(4)

Since $\frac{d}{dt} \left[ \sum_{i=0}^{k} \alpha_i \xi^i \right] \neq 0$ because $\xi = 1$ is a simple root, it follows from eq. (4) that $\sum_{i=0}^{k} \beta_i \neq 0$. 


Lemma 1. If the following three hypotheses are true:

1) \( \sum_{i=0}^{\infty} a_i Z^i \)

2) \( b_0 = 1, b_i > 0 \) for \( i = 1, 2, \ldots \)

3) \( b_{n+1} b_{n-1} - b_n > 0 \)

then \( a_0 = 1 \) and \( a_i < 0 \) for \( i = 1, 2, \ldots \)

Proof:

If \( \sum_{i=0}^{\infty} b_i Z^i \) has a non-zero radius of convergence, then \( \sum_{i=0}^{\infty} a_i Z^i \) does also.

(See Knopp, K. Infinite Sequences and Series, Dover, 1956, p. 116.)

From

\[ \sum_{i=0}^{\infty} b_i Z^i \cdot \sum_{i=0}^{\infty} a_i Z^i = 1 \]

we obtain the following:

\[ b_0 a_0 = a_0 = 1 \]
\[ b_1 + a_1 = 0 \]
\[ b_2 + b_1 a_1 + a_2 = 0 \]
\[ \vdots \]
\[ b_n + b_{n-1} a_1 + \ldots + b_1 a_{n-1} + a_n = 0 \]

Solving for \( b_n \):

\[ b_n = -\sum_{j=1}^{n} b_{n-j} a_j \]  \hspace{1cm} (5)

In a similar manner

\[ b_{n+1} = -\sum_{j=1}^{n} b_{n+1-j} a_j - a_{n+1} \]  \hspace{1cm} (6)

Multiplying eq. (6) by \( b_n \) and eq. (5) by \( b_{n+1} \) and subtracting gives

\[ b_n a_{n+1} = \sum_{j=1}^{n} (b_{n-j} b_{n+1} - b_{n+1-j} b_n) a_j \]  \hspace{1cm} (7)
Since by hypothesis \( b_{n+1} b_{n-1} - b_n^2 > 0 \),

\[
\begin{align*}
\frac{b_0}{b_1} > \frac{b_1}{b_2} > \frac{b_2}{b_3} > \cdots > \frac{b_{n-j}}{b_{n+1-j}} > \cdots > \frac{b_n}{b_{n+1}}
\end{align*}
\]

\( b_{n-j} b_{n+1} - b_{n+1-j} b_n > 0 \) and this shows the term in parenthesis in eq. (7) is always positive. Equation (7) will now be used to show by induction that \( a_1 < 0 \), \( i - 1 \), \( 2, \ldots \)

\[ a_0 = 1 \]

\[ a_1 = -b_1 < 0 \]

from eq. (7)

\[ b_1 a_2 = (b_0 b_2 - b_1^2) a_1 < 0 \]

\[ \ldots a_2 < 0 \]

Assume \( a_1, a_2, \ldots a_n < 0 \). From eq. (7)

\[ a_{n+1} = \frac{1}{b_n} \sum_{j=1}^{n} \left( b_{n-j} b_{n+1} - b_{n+1-j} b_n \right) a_j \]

Since all the terms in parentheses are positive, \( b_n \) is positive, and the \( a_j \)'s are negative, \( a_{n+1} \) must be negative. This completes the induction.

Theorem: No explicit linear multistep method remains stable as \( h \) becomes arbitrarily large. \( p = k \) is the highest order possible for an implicit linear multistep method which remains stable as \( h \) becomes arbitrarily large.

Proof: let \( \dot{y} = qh \) (8)

The linear multistep method used to solve eq. (8) can be written as:

\[ \sum_{i=0}^{k} a_i y_{n+1} - h \sum_{i=0}^{k} \beta_i y_{n+i} = O(h^{p+1}) \] (9)

Substituting the solution \( y = e^{qt} \) of eq. (8) into eq. (9):

\[ \sum_{i=0}^{k} a_i (e^{qh})^{n+i} - qh \beta_1 (e^{qh})^{n+1} = O(h^{p+1}) \] (10)

Let us use the transformation \( \zeta = e^{qh} \) which maps the LHP of \( qh \) into the unit circle in the \( \zeta \) plane. Dividing eq. (10) by \( qh \) and substituting \( \zeta = e^{qh} \) and \( qh = \ln \zeta \):

\[ \sum_{i=0}^{k} \frac{a_i}{\ln \left( \frac{1+Z_i}{1-\zeta} \right)} - \sum_{i=0}^{k} \beta_1 \ln \zeta = O \left( |\ln \zeta|^p \right) \] (11)
The $\xi$'s may be viewed as the roots of eq. (9). For stability they must remain in the unit circle. The transformation $Z = \frac{\xi - 1}{\xi + 1}$ maps the interior of the unit circle into the LHP of the Z plane. The inverse relationship is

$$\frac{\xi - 1}{\xi + 1}$$

Substituting eq. (12) into eq. (11)

$$\frac{k}{\ln \frac{1+z}{1-z}} = \sum_{i=0}^{k} a_i \left( \frac{1+z}{1-z} \right)^i$$

As $h \to 0$, $\xi \to 1$, and $Z \to 0$. Therefore multiplying eq. (13) by $\frac{1-Z}{Z}$ and simplifying:

$$\frac{r(Z)}{\ln \frac{1+z}{1-z}} = S(Z) = O(Z^p)$$

where

$$r(Z) = \frac{1-Z}{Z} \sum_{i=0}^{k} a_i \left( \frac{1+z}{1-z} \right)^i = \sum_{i=0}^{k} a_i Z^i$$

$$S(Z) = \frac{1-Z}{Z} \sum_{i=0}^{k} \beta_i \left( \frac{1+z}{1-z} \right)^i = \sum_{i=0}^{k} b_i Z^i$$

If eq. (14) is to be of order $p$ then the coefficients on the left side of eq. (14) must cancel up to the $Z^p$ term. $r(Z)$ has a zero at $Z = 0$ because $\sum_{i=0}^{k} a_i \xi^i$ has a root at $\xi = 1$. Since eq. (9) is stable for $h = 0$ all the roots of $\sum_{i=0}^{k} a_i \xi^i$ must be on or within the unit circle and hence $a_i \geq 0$ in eq. (15). If we desire that all the roots of eq. (9) be on or within the unit circle as $h \to \infty$ then $b_i \geq 0$ also. Let

$$\frac{Z}{\ln \frac{1+z}{1-z}} = C_0 + C_2 Z^2 + C_4 Z^4 + \ldots$$

Due to Lemma 1:

$$C_0 > 0, \ C_2, \nu < 0, \ \nu = 1, 2, \ldots$$

Matching coefficients in eq. (14) gives

$$b_0 = c_0 a_1$$

$$b_1 = c_0 a_2$$

(18)
\[ b_{2v} = c_0 a_{2v+1} + c_2 a_{2v-1} + \cdots + c_2 a_{1} \]
\[ b_{2v+1} = c_0 a_{2v+2} + c_2 a_{2v} + \cdots + c_2 a_{2} \]

Since \( S(Z) \) is a polynomial of \( k \)th order \( b_n \neq 0, n > k \). The same applies for \( a_n \neq 0, n > k \).

Assume \( k \) is odd.

\[ b_{2v+1} = b_k = c_0 a_{k+1} + c_2 a_{k-1} + \cdots + c_2 a_{2} \]

Since \( a_{k+1} < 0 \), the coefficient matching in eq. (14) requires that \( b_k < 0 \). But by Descartes' Rule of Signs this means that \( S(Z) \) has roots in the right half of the \( Z \) plane and hence \( \| Z \| > 1 \) as \( h \to \infty \). The conclusion is that we can only require coefficient matching in eq. (14) up to the \( k-1 \) th term. Hence \( p = k \). The same reasoning holds if \( k \) is even.

If eq. (9) is explicit, then \( \beta_k = 0 \). From eq. (16) \( S(l) = 0 = \sum_{i=0}^{k} \beta_i \). Since \( \beta_i \geq 0, i = 0, \ldots, k \) if all the roots are to be within the unit circle as \( h \to \infty \), all the \( \beta_i \)'s must be zero, including \( \beta_0 \). But \( \beta_0 = (\frac{k}{2}) \sum_{i=0}^{k} \beta_i \) from eq. (16).

From the consistency condition

\[ \sum_{i=0}^{k} \beta_i \neq 0 \]

Hence if \( \beta_k = 0 \) and if we require stability as \( h \to \infty \), we find the consistency condition is violated. The conclusion is that no explicit linear multistep method can be stable as \( h \to \infty \).
GEAR'S FORMULA

\( k = 2 \)

\[ y_{n+1} = \frac{4}{3} y_n - \frac{1}{3} y_{n-1} + h \left( \frac{2}{3} y_{n+1} \right) \]

\( k = 3 \)

\[ y_{n+1} = \frac{18}{11} y_n - \frac{9}{11} y_{n-1} + \frac{2}{11} y_{n-2} + h \left( \frac{6}{11} y_{n+1} \right) \]

\( k = 4 \)

\[ y_{n+1} = \frac{48}{25} y_n - \frac{36}{25} y_{n-1} + \frac{16}{25} y_{n-2} - \frac{3}{25} y_{n-3} + h \left( \frac{12}{25} y_{n+1} \right) \]

\( k = 5 \)

\[ y_{n+1} = \frac{300}{137} y_n - \frac{300}{137} y_{n-1} + \frac{200}{137} y_{n-2} - \frac{75}{137} y_{n-3} + \frac{12}{137} y_{n-4} + h \left( \frac{60}{137} y_{n+1} \right) \]

\( k = 6 \)

\[ y_{n+1} = \frac{360}{147} y_n - \frac{450}{147} y_{n-1} + \frac{400}{147} y_{n-2} - \frac{225}{147} y_{n-3} + \frac{72}{147} y_{n-4} - \frac{10}{147} y_{n-5} + h \left( \frac{60}{147} y_{n+1} \right) \]

Coefficients of Optimized Formula (k = 4):

\[ \alpha_0 = 1.584 \]
\[ \alpha_1 = -1.017 \]
\[ \alpha_2 = 0.529 \]
\[ \alpha_3 = -0.0968 \]
\[ \beta_0 = 0.235 \]
\[ \beta_1 = 0.0568 \]
\[ \beta_2 = 0.00567 \]
\[ \beta_3 = 0.000201 \]
TRUNCATION ERROR

An error term for a Taylor series expansion of \( y(t) \) about a point in time at \( nh \) can be obtained from the integral

\[
\frac{1}{(n-i)h} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^{(r+1)} y^{(r+1)}(s) \, ds
\]

If this is integrated by parts, we obtain

\[
\frac{1}{(n-i)h} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^r y^{(r+1)}(s) \, ds = \frac{1}{(n-i)h} \left( [n-i]h-s \right)^r y^{(r)}(s) \bigg|_{nh}^{(n-i+1)h} 
\]

\[
+ \frac{1}{(r-1)!} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^{(r-1)} y^{(r)}(s) \, ds
\]

\[
= -\left( \frac{(-h)^r}{r!} y^{(r)}(nh) + \frac{1}{(r-1)!} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^{(r-1)} y^{(r)}(s) \, ds \right)
\]

If integration by parts is carried out \( r \) times and the terms rearranged:

\[
y([n-i]h) = y(nh) - ih y^{(1)}(nh) + \frac{h^2}{2!} y^{(2)}(nh) - \ldots
\]

\[
+ \frac{(-1)^{r-1} h^r}{r!} y^{(r)}(nh) + \frac{1}{(r-1)!} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^{(r-1)} y^{(r)}(s) \, ds
\]

A Taylor series expansion with error term can be obtained in the same manner for the first derivative of \( y(t) \) about a point \( nh \).

\[
y^{(1)}([n-i]h) = y^{(1)}(nh) - ih y^{(2)}(nh) + \ldots
\]

\[
+ \frac{(-1)^{r-1} h^r}{(r-1)!} y^{(r)}(nh) + \frac{1}{(r-1)!} \int_{nh}^{(n-i+1)h} ([n-i]h-s)^{(r-1)} y^{(r+1)}(s) \, ds
\]

Equations (1) and (2) can be substituted into the integration formula in eq. (3) and an expression for \( T_n \) can be obtained.

\[
y_{n+1} = \sum_{i=0}^{k-1} \alpha_i y_{n-i} + h \sum_{i=-1}^{k-1} \beta_i \dot{y}_{n-i} + T_n
\]

Since eq. (3) is exact up to and including order \( p \) we obtain for \( T_n \):
\[ T_n = \frac{1}{p!} \int_{(n+1-k)h}^{(n+1)h} \left\{ \left(\frac{(n+1)}{h-s}\right)^P - \phi \beta_{-1} \left(\frac{(n+1)}{h-s}\right)^{r-1} \right\} y^{(p+1)}(s) \, ds \]

where

\[ \left(\frac{(n-i)}{h-s}\right) = \begin{cases} 
\frac{(n-i)}{h-s}, & [n-i]h \leq s \leq nh, \quad i \neq 1 \\
0, & \text{otherwise}
\end{cases} \]

Equation (4) can be written

\[ T_n = \frac{1}{p!} \int_{(n+1-k)h}^{(n+1)h} G(s) y^{(p+1)}(s) \, ds \]  

(5)

If \( G(s) \) is of the same sign over the interval \([n+1-k)h, (n+1)h]\), then \( \int G(s) \, ds \) is a monotonic function. If \( y^{p+1}(s) \) is continuous over the interval, then the First Mean Value Theorem applies. The estimate for the truncation error can be written:

\[ T_n = \frac{1}{p!} y^{p+1}(\eta) \int_{(n+1-k)h}^{(n+1)h} G(s) \, ds \]

(6)

where \( \eta \in (n+1-k)h, (n+1)h \).

Example

Let \( k = p = 4 \)

\[ y_{n+1} = \alpha \, y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \alpha_3 y_{n-3} + \]

\[ + h \left\{ \beta_{-1} \dot{y}_{n+1} + \beta_0 \dot{y}_n + \beta_1 \dot{y}_{n-1} + \beta_2 \dot{y}_{n-2} + \beta_3 \dot{y}_{n-3} \right\} + T_n \]

(7)

The influence function is obtained from eq. (4).

\[ G(s) = G_1(s) = \alpha_3 \left(\frac{|n-3|}{h-s}\right)^4 + 4h\beta_3 \left(\frac{|n-3|}{h-s}\right) \]

for \( |n-3|h \leq s \leq |n-2|h \)
\[ G(s) = G_2(s) = G_1(s) + \alpha_2 ([n-2|h-s])^4 + 4h\beta_2 ([n-2|h-s])^3 \]

for \(|n-2|h \leq s \leq |n-1|h\)

\[ G(s) = G_3(s) = G_2(s) + \alpha_1 ([n-1|h-s])^4 + 4h\beta_1 ([n-1|h-s])^3 \]

for \(|n-1|h \leq s \leq nh\)

\[ G(s) = ([n+1|h-s])^4 - 4h\beta_{-1} ([n+1|h-s])^3 \]

for \(nh \leq s \leq [n+1]h\)

For Gear's Method

\[ \alpha_0 = \frac{48}{25}, \alpha_1 = -\frac{36}{25}, \alpha_2 = \frac{16}{25}, \alpha_3 = -\frac{3}{25}, \beta_{-1} = \frac{12}{25}, \]

\[ \beta_0 = \beta_1 = \beta_2 = \beta_3 = 0 \]

The influence function is shown in Figure 1. Since the influence function is of the same sign over the interval \([n-3|h, [n+1|h]\), the error is

\[ T_n = \frac{E_4}{5^5} \cdot y^{(5)}(\eta) h^5 \]

To find \(E_4\) assume \(y(t) = t^5\). Substituting this into the integration formula:

\[ h^5 = -\alpha_1 h^5 - 32\alpha_2 h^5 - 243\alpha_3 h^5 + 5h^5 (\beta_{-1} + \beta_1 + 16\beta_2 + 81\beta_3) \]

+ \(E_4 h^5\)

Solving for \(E_4\):

\[ E_4 = 1 + \alpha_1 + 32\alpha_2 + 243\alpha_3 - 5 (\beta_{-1} + \beta_1 + 16\beta_2 + 81\beta_3) \]

Using the values of the \(\alpha's\) and \(\beta's\) for Gear's Method gives

\[ E_4 = -11.52 \left| T_n \right| = \frac{2.304}{4^5} h^5 \left| y^{(5)}(\eta) \right| \]

This shows that

\[ \left| \int_{(n-3)h}^{(n+1)h} G(s) \, ds \right| = 2.304 h^5 \]

which can be verified by integrating the influence function directly.
APPENDIX B

SOURCE LISTING

FLOW CHART OF PROGRAM

SAMPLE PROBLEM
*COMMON Y0,DY0,D5,T,H,NV,NUM,Y1,Y2,Y3,Y4,E1,E2,DY1,DY2,DY3, 
1 Dy4,TRR,DT,INDEX,JAC,G,S5, 
COMMON BND,S,YLIM,XNLIM,ELIM,SC,NEQ3,NH 
DOUBLE PRECISION YO(10),DYO(10),DB(10),Y1(10),Y2(10),Y3(10) 
Y4(10),O(10)+G(10)+10, 
2 YS(i10),Y6(i10),Y7(i10),DY5(i10),DY6(i10),DY7(i10) 
DOUBLE PRECISION X,OMAX,SET 
DATA LIM/9/,HM/S/1*DE-15/ 
C INITIALIZE NF COUNTS STARTING FAILURES 
C N4 COUNTS SUCCESSFUL INTEGRATIONS 
C N6 COUNTS STEPS SINCE LAST HALVING OR DOUBLING 
C N9 FLAGS CONVERGENCE IN FPD 
C LIMITS THE STARTING FAILURE LIMIT 
C JAC COUNTS MATRIX INVERSIONS = JACOBIAN EVALUATION 
C NUM COUNTS EVALUATIONS OF STATE EQUATIONS 
3 FORMAT(2HI,INITIAL CONDITIONS) 
ASSIGN 98 TO NSW 
NH = 0 
NF = 0 
JAC = 0 
NUM = 0 
READ 5,TF,PT,NV 
NEQ3 = MAXO(NV*3,7) 
INDEX = 6 
DT = TF/PT 
TRR = DT 
H = TF * 0.001 
READ 6,LYO(I),I=1,NV) 
6 FORMAT(10F8.2) 
PRINT 4,(YO(I),I=1,NV) 
FORMATT(1X,6E15.7) 
READ 101YLYM,ELIM,XNLIM,SC 
101 FORMAT(4E7.1) 
PRINT 100 
100 FORMAT(9HO LIMITS) 
PRINT 4,(YLIM,ELIM,XNLIM,SC 
ELIM = ELIM * SC 
EDUB = ELIM /40 
CALL STATE(YO,DY3) 
DO 261 I=1,NV 
261 Y3(I) = YO(I) 
CALL GILL 
USE GILL'S METHOD TO START 
10 -T = 0 
N4 = 3 
N6 = 3 
CALL GILL 
DO 262 I=1,NV 
262 Y2(I) = YO(I) 
CALL GILL 
DO 263 I=1,NV 
263 Y1(I) = YO(I) 
CALL GILL 
C
INITIALLY, THE 4 POINT SCHEME IS USED

DO 110 I = 1, NV

110 Y(I) = 4*(Y0(I) + Y2(I)) - 6*Y1(I) - Y3(I)

T = T + H

CALCULATE INITIAL JACOBIAN

CALL STATE (Y, DY)

DO 13 I = 1, NV

X = Y(I)

DIF = DABS(X) - YLIM

IF (DIF) 16, 16, 16

16 Y(I) = X + 1.0D-9

SET = 1

GO TO 19

19

15 Y(I) = X + 1.0D-9

SET = 1000 * X

GO TO 18

CALL STATE (Y, DY)

CALL OX(G, QMAX)

DO 14 J = 1, NV

G(J, I) = (Q1(J) - DB(J)) * SET

CALL MATINV (NV, G)

JAC = JAC + 1

GO TO 107

TO HERE WHEN H IS TO BE DOUBLED

-222 H = H + H

N6 = 4

SHIFT Y'S = Y1 IS ALREADY IN POSITION

DO 180 I = 1, NV

DY0(I) = DY(I)

DY2(I) = DY3(I)

DY3(I) = DY5(I)

DY4(I) = DY7(I)

Y0(I) = Y(I)

Y2(I) = Y3(I)

Y3(I) = Y5(I)

180

Y4(I) = Y7(I)

USE 5 POINT FORMULA ONCE STARTED

DO 111 I = 1, NV

Y(I) = 5*(Y0(I) + 2*Y2(I) - Y1(I) - Y3(I)) + Y4(I)

T = T + H

CALL FPD

IF (NH) 305, 305, 305

305 FORMAT(3HHW12.5H

N5 = 14.5H

N6 = 15.5H

N4 = 15)

IF (N6 - 5) 500, 450, 450

202 ASSIGN 99 TO N5

GOOD POINT—DOWNSHIFT AND MOVE ON

DO 120 I = 1, NV

Y7(I) = Y6(I)
**Language:** English  
**Content:**

119. \( Y6(1) = Y5(1) \)
120. \( Y5(1) = Y4(1) \)
121. \( Y4(1) = Y3(1) \)
122. \( Y3(1) = Y2(1) \)
123. \( Y2(1) = Y1(1) \)
124. \( Y1(1) = Y0(1) \)
125. \( Y0(1) = Y(1) \)
126. \( D7(1) = D8(1) \)
127. \( D6(1) = D5(1) \)
128. \( D5(1) = D4(1) \)
129. \( D4(1) = D3(1) \)
130. \( D3(1) = D2(1) \)
131. \( D2(1) = D1(1) \)
132. \( D1(1) = D0(1) \)
133. \( D0(1) = D(1) \)
134. \( \text{GO TO 105} \)
135. \( \text{C} \)
136. \( \text{C} \)
137. \( \text{C} \)
138. \( \text{C} \)
139. \( \text{C} \)
140. \( \text{C} \)
141. \( \text{C} \)
142. \( \text{C} \)
143. \( \text{C} \)
144. \( \text{C} \)
145. \( \text{C} \)
146. \( \text{C} \)
147. \( \text{C} \)
148. \( \text{C} \)
149. \( \text{C} \)
150. \( \text{C} \)
151. \( \text{C} \)
152. \( \text{C} \)
153. \( \text{C} \)
154. \( \text{C} \)
155. \( \text{C} \)
156. \( \text{C} \)
157. \( \text{C} \)
158. \( \text{C} \)
159. \( \text{C} \)
160. \( \text{C} \)
161. \( \text{C} \)
162. \( \text{C} \)
163. \( \text{C} \)
164. \( \text{C} \)
165. \( \text{C} \)
166. \( \text{C} \)
167. \( \text{C} \)
168. \( \text{C} \)
169. \( \text{C} \)
170. \( \text{C} \)
171. \( \text{C} \)
172. \( \text{C} \)
173. \( \text{C} \)
174. \( \text{C} \)
175. \( \text{C} \)
176. \( \text{C} \)

---

**New Content:**

119. \( Y6(1) = Y5(1) \)
120. \( Y5(1) = Y4(1) \)
121. \( Y4(1) = Y3(1) \)
122. \( Y3(1) = Y2(1) \)
123. \( Y2(1) = Y1(1) \)
124. \( Y1(1) = Y0(1) \)
125. \( Y0(1) = Y(1) \)
126. \( D7(1) = D8(1) \)
127. \( D6(1) = D5(1) \)
128. \( D5(1) = D4(1) \)
129. \( D4(1) = D3(1) \)
130. \( D3(1) = D2(1) \)
131. \( D2(1) = D1(1) \)
132. \( D1(1) = D0(1) \)
133. \( D0(1) = D(1) \)
134. \( \text{GO TO 105} \)
135. \( \text{C} \)
136. \( \text{C} \)
137. \( \text{C} \)
138. \( \text{C} \)
139. \( \text{C} \)
140. \( \text{C} \)
141. \( \text{C} \)
142. \( \text{C} \)
143. \( \text{C} \)
144. \( \text{C} \)
145. \( \text{C} \)
146. \( \text{C} \)
147. \( \text{C} \)
148. \( \text{C} \)
149. \( \text{C} \)
150. \( \text{C} \)
151. \( \text{C} \)
152. \( \text{C} \)
153. \( \text{C} \)
154. \( \text{C} \)
155. \( \text{C} \)
156. \( \text{C} \)
157. \( \text{C} \)
158. \( \text{C} \)
159. \( \text{C} \)
160. \( \text{C} \)
161. \( \text{C} \)
162. \( \text{C} \)
163. \( \text{C} \)
164. \( \text{C} \)
165. \( \text{C} \)
166. \( \text{C} \)
167. \( \text{C} \)
168. \( \text{C} \)
169. \( \text{C} \)
170. \( \text{C} \)
171. \( \text{C} \)
172. \( \text{C} \)
173. \( \text{C} \)
174. \( \text{C} \)
175. \( \text{C} \)
176. \( \text{C} \)

---

**New Content:**

119. \( Y6(1) = Y5(1) \)
120. \( Y5(1) = Y4(1) \)
121. \( Y4(1) = Y3(1) \)
122. \( Y3(1) = Y2(1) \)
123. \( Y2(1) = Y1(1) \)
124. \( Y1(1) = Y0(1) \)
125. \( Y0(1) = Y(1) \)
126. \( D7(1) = D8(1) \)
127. \( D6(1) = D5(1) \)
128. \( D5(1) = D4(1) \)
129. \( D4(1) = D3(1) \)
130. \( D3(1) = D2(1) \)
131. \( D2(1) = D1(1) \)
132. \( D1(1) = D0(1) \)
133. \( D0(1) = D(1) \)
134. \( \text{GO TO 105} \)
135. \( \text{C} \)
136. \( \text{C} \)
137. \( \text{C} \)
138. \( \text{C} \)
139. \( \text{C} \)
140. \( \text{C} \)
141. \( \text{C} \)
142. \( \text{C} \)
143. \( \text{C} \)
144. \( \text{C} \)
145. \( \text{C} \)
146. \( \text{C} \)
147. \( \text{C} \)
148. \( \text{C} \)
149. \( \text{C} \)
150. \( \text{C} \)
151. \( \text{C} \)
152. \( \text{C} \)
153. \( \text{C} \)
154. \( \text{C} \)
155. \( \text{C} \)
156. \( \text{C} \)
157. \( \text{C} \)
158. \( \text{C} \)
159. \( \text{C} \)
160. \( \text{C} \)
161. \( \text{C} \)
162. \( \text{C} \)
163. \( \text{C} \)
164. \( \text{C} \)
165. \( \text{C} \)
166. \( \text{C} \)
167. \( \text{C} \)
168. \( \text{C} \)
169. \( \text{C} \)
170. \( \text{C} \)
171. \( \text{C} \)
172. \( \text{C} \)
173. \( \text{C} \)
174. \( \text{C} \)
175. \( \text{C} \)
176. \( \text{C} \)
177*    DY4(1) = DY2(1)
179*    DY2(1) = DY1(1)
180*    YZ(1) = YMAX
182*    T = T+H+H:
183*    CALL STATE (Y1, DY1)
184*    T = T+H
185*    GO TO 105
187*    C    ERROR IN BOUNDS - CHECK WHETHER TO PRINT
188*    212 IF (T-TPR) .LT.0.301,301
189*    301 CALL PRN
190*    IF (TPR-IF) .GT.212,212,304
191*    304 IF (T-TF) .GT.302,302,306
192*    C    CHECK WHETHER TO DOUBLE STEP SIZE - DELAY OF 3 PASSES
194*    C    IS MANDATED BY STRUCTURE OF POINT-SAVING MECHANISM
195*    302 IF (N6-N) 500,480,480
196*    480 IF (E2=EDUB),222,222,500
197*    991 PRINT 1, NF, E1
198*    1 FORMAT(1HO,12,28H STARTING FAILURES = ERROR =E15+7)
199*    990 PRINT 2, H, T
200*    2 FORMAT(21HSTEP TOO SMALL = H =E15+7, 4H T =E15+7)
201*    305 PRINT 78, (-1, Y(I)), Y1(I), Y2(I), Y3(I), Y4(I), Y5(I), Y6(I)), I = 1, NV)
202*    78 FORMAT(1HO,16F17,8)
203*    306 PRINT 7, NUM
204*    7 FORMAT(12H STATE EQUATIONS EVALUATED; 15,16 H TIMES)
205*    8 PRINT 8, JAC
206*    8 FORMAT(19H JACOBIAN EVALUATED; 15,16 H TIMES)
207*    STOP
208*    END

END OF COMPILED:
NO DIAGNOSTICS.
SUBROUTINE FPD
COMMON Y0,Y1,Y2,Y3,Y4,E1,E2,DY,DY1,DY2,DY3,DY4,DY5,DY6,DY7,DY8,DY9
COMMON BND5/YLIM,XLIM,ELIM,SC,NEQ3,NH
DOUBLE PRECISION Y0,Y1,Y2,Y3,Y4,E1,E2,DY,DY1,DY2,DY3,DY4,DY5,DY6,DY7,DY8,DY9
DOUBLE PRECISION XQMAX,SET,SUM,DIV,EQUIVALENCE (NV,NVZ),(Y0,Y1),(YLIM,YLIM),(OLDB,EXTRA)
DATA NAT/6,NHLIM/I7/
N5=COUNTS EVALUATIONS OF R
N1=NO FLAGS TWO SUCCESSIVE INDEX STEP FAILURES
N4=COUNTS THESE INDEX STEPS
N5=1
N1=1
ASSIGN 85 TO NHX
CALL STATE (Y,DY)
CALL QX(DQ,QMAX)
PI1=0.
DO 20 J=1,NV2
PI1=PI1+QX(DQ(J)/QMAX)**2
PI1 = SQRT(PI1)*QMAX.
40 DO 163 I=1,NV
CURDY(I) = DY(I)
163 OBD(I) = B(I)
50 N6 = 1
IOUT = NV
DO 25 I=1,NV2
75 SUM=O.DO
DO 35 J=1,NV2
35 SUM = SUM + (X(I,J))*DQ(I,J)
37 IFABS(SUM-NHLIHM/SC+ABS(OLDB(I)))>199,199,25
38 IOUT = IOUT+1
39 DO 25 J=1,NV2
40 IF ALL IN BOUNDS, PRESENT R'S ARE THE SOLUTION
41 IF (IOUT-2.10+210+67)
67 BN = 0.
42 DBN = 0.
43 DO 68 I=1,NV
68 BN = -BN + 8(I) + B(I)
45 DBN = DBN + DB(I) + DB(I)
47 DF = (DBN-BN)/63,63,64
48 TI = 1.
49 TI = SQRT(BN/DBN)
52 GO TO 60
50 GO TO 60
51 Ti = SQRT(BN/DBN)
52 GO TO 60
53 DO 205 I=1,NV
54 DO 205 I=1,NV
55 NH = 0
56 RETURN
57 IF NORM OF DB TOO LARGE, RECALCULATE R AND GET NEW PI
58 Ti = 1.
60 DO 55 I=1,NV2.
IF INDEX FAILURES, CALCULATE NEW G = HALVE IF THIS FAILS

IF (N10) 230, 230, 220

230 PRINT '77, N5, I1
77 FORMAT (26HONDO CONVERGENCE, N10=0, N5=, I4, 4H H =, E15.7, 4H T =, E15.7)

305 PRINT '77, (Y(I), Y(J), Y(K))
78 FORMAT (1X, 6 E16.8)

IF (NH-NHlim) 575, 576, 576

575 NH = NH+1
576 RETURN

RETURN

220 N10=0

C MUST RESTORE OLD B'S

DO 161 J=1, NV
161 B(I) = OLDB(I)

C FIND G

DO 13 I=1, NV2
13 X = B(I)

DIF = DABS(X) - BLIM
IF (DIF) 16, 16, 15

16 B(I) = X = I, DO=4

SET = I*DO

GO TO 18

18 CALL STATE (Y, DY)

CALL G*Q, GHAX

DO 14 J=1, NV2
14 G(J, I) = Q(J, I) = DQ(J) + SET

G(J, I) = X

CALL MATHINV (NV2, G)

JAC = JAC+1

C ITERATE ON THE RESIDUAL

DO 302 H=1, LIM
302 DO 415 I=1, NV
415 SUM = 0*DO

105 DO 412 J=1, NV
106 SUM = 0*DO
107 DO 413 K=1, NV
108 SUM = SUM + SAVE(I, K) * G(K, J)
109 RES(I, J) = -SUM
110 RES(I, I) = 2*DO + RES(I, I)
111 DO 420 J=1, NV
112 DO 417 J=1, NV
113 SUM = 0*DO
114 DO 418 K=1, NV
115 SUM = SUM + G(I, K) * RES(K, J)
116 DB(I, J) = SUM
117 DO 419 J=1, NV
118 DO 419 G(I, J) = DB(I, J)
119 CONTINUE
120 CONTINUE
121 GO TO 50
122 C TO HERE IF LESS THAN NDFX FAILURES
124 CALL STATE (Y, DY)
125 CALL Q(X, QMAX)
126 P12 = 0
127 DO RO I = 1, NV2
128 P12 = P12 + ((4*Q(I)/QMAX) ** 2
129 P12 = SQRT(P12)*QMAX
130 C CHECK NEW P
131 C IF(P12-P11) 87, 90, 90
133 87 GO TO NHEX(85, 86)
134 IF(N5=200) 240, 250, 250
135 250 PRINT 260, HT
136 FORMATTED CONVERGENCE, N5=200, H=15, 7, H=15, 7
137 GO TO 305
138 85, IF(N5=NEQ3) 240, 240, 245
139 245 ASSIGN 86 TO NHEX
140 GO TO 75
141 C ITERATE TO GET NEW G AND TRY AGAIN
143 P11 = P12
144 NI0 = 1
145 DO 110 I = 1, NV2
146 DQ(J) = Q(J) + DQ(J)
147 DIV = 0 DD
148 DO 150 I = 1, NV2
149 SUM = 0 DD
150 DO 151 J = 1, NV2
151 SUM = SUM + DQ(I)*G(I, J)
152 DIV = DIV + DB(J) + SUM
153 EXTRA(I) = SUM * T1 * DB(I)
154 DIV = DIV - DIV
155 DO 155 I = 1, NV2
156 SUM = EXTRA(I) - DIV
157 DO 154 J = 1, NV2
158 SAVE(I, J) = DB(J) + I + SUM
159 SAVE(I, I) = 1, *SAVE(I, I)
160 DO 156 K = 1, NV2
161 DO 157 I = 1, NV2
162 SUM = 0 DD
163 DO 158 J = 1, NV2
164 SUM = SUM + SAVE(J) + G(J, K)
165 EXTRA(I) = SUM
166 DO 159 I = 1, NV2
167 G(I, K) = EXTRA(I)
168 DO 156 J = 1, NV2
169 GO TO 40
170 END

END OF COMPILATION: NO DIAGNOSTICS.
SUBROUTINE PRYNT

COMMON Y0,DY0,DD,T,TN,Y1,Y2,Y3,Y4,E1,E2,DY,Y1,DY2,DY3,

1  DY4,TPR,DT

DOUBLE PRECISION Y0(10),DY0(10),DD(10),Y1(10),Y2(10),Y3(10),

1  Y4(10),DY1(10),DY2(10),DY3(10),DY4(10)

DIMENSION V(4),Z(10)

EQUIVALECE (Z,DB)

LAGRANGIAN INTERPOLATION TO MAKE THE TIME COME OUT NICE

A1 = T

DO 10 J=1,6

10 PROD = PROD * 1

DO 12 K=1,6

12 D = T

IF(K-J) 20,12,20

20 PROD = PROD * (TPR+DT)/(A1-D)

12 D = D-H

10 PROD = V(J)

10 A1 = A1-H

DO 30 J=1,NV

30 Z(J) = V(1)*Y(J)+V(2)*Y(J)+V(3)*Y(J)+V(4)*Y(J)

11 V(6)*Y(J)

PRINT 40,TPR

FORMAT (4H6T,E15.7)

PRINT 60,(Z(J),I=1,NV)

FORMAT (2X,5G14.6)

RETURN

END
SUBROUTINE GILL

COMMON YO,DYO,U,T,NV,NUM

DOUBLE PRECISION YO(10),DYO(10),U(10),A(4,4)

DATA A/5D00.,292893218813452476D0.,1.70710678118654752D0,1.292893218813452476D0.

DOUBLE PRECISION YD(10),DYO(10),U(10),A(4,4)

COMMON YODYO,THNV,NUM

DATA A/95D0,.292893218813q52476Do1)707106781 1865475200,.166666666

CALL STATE(YODYO)

DO 500 J=1,NV

500 U(J)=0.D0

J=1

DO 1040 J=1,NV

1040 U(J) = A(J,1) * DY0(J) - A(J,2) * U(J)

GO TO (1060, 1060, 1060, 950) + J

1060 T=T+H=5

1010 J=J-1

GO TO 1040

950 CALL STATE(YODYO)

RETURN

END

END OF COMPIlATION: NO DIAGNOSTICS.

SUBROUTINE DSY

COMMON YO,DYO,NUM,Y,Y2,Y3,Y4,E1,E2

DOUBLE PRECISION YO(10),DYO(10),NUM,Y,Y2,Y3,Y4,E1,E2

1 YD(10)

COMMON ANDS,YLIM,XHLM,ELIM

DATA DLIM/*0.05/

C CHOOSE MAXIMUM TRUNCATION ERROR (SORT OF)

E1=-1.0E20

E2 = E1

DO 20 J=1,NV

ER = ABS(Y(J)) + 5.0 * (Y3(J)+2.0 * (Y(J)-Y2(J))) - YO(J) + Y4(J)) * DLIM

ES=ABS(Y(J)) * ELIM

ET = ER + ES

IF(ET=ET1) GO TO 30

30 E1 = ET

10 ET=ET+ES/40.

IF(ET<=ET1) GO TO 20,20

GO TO 40

CONTINUE

RETURN

END

END OF COMPIlATION: NO DIAGNOSTICS.
SUBROUTINE OX(QZ)
COMON#-e,-tBiP'TIN,
WvNUM,fltY

DOUBLE PRECISION YOIO),DYO0(I)ODBlO)IyCO)IYIcIIo)tY2(I0) Y3(10)

DOUBLE PRECISION QZ.X

DATA A1/-1 ,016965299999999A671DO/
DATA A2/-529444049999979880D0/
DATA A3/-96778749999910463D-1/
DATA BM/459866922174713980D0/
DATA BO/23502630946787243D0/
DATA B1/566199765486269340D-1/
DATA B2/567200528010787490D-2/
DATA B3/-20136700809476272D-3/

C C

DATA C/-1 ,016965299999999A671DO/
DATA C/-529444049999979880D0/
DATA C/-96778749999910463D-1/
DATA C/-459866922174713980D0/
DATA C/-23502630946787243D0/
DATA C/-566199765486269340D-1/
DATA C/-567200528010787490D-2/
DATA C/-20136700809476272D-3/

CMATINV

SUBROUTINE CMATINV(N,Z)
DOUBLE PRECISION Z,X,CC

DIMENSION IF(I),Z(I+1),X(I+1),Y(I+1),2,1X(10,2)

DO 10 I=1,N
10 P(I)=0

DO 76 I=1,N
76 X=O+D0

DO 69 J=1,N
69 X=CC

IF(IF(J)+1).LT.41,69,61
61 DO 68 K=1,N
68 CC=DA85(Z(J,K))

IF(X+CC).LT.67,68,68
68 12=J

12=K

X=CC

68 CONTINUE.

69 CONTINUE.

IF(11+1).LT.11+1

IF(12-11).GT.70,72,70

70 DO 71 L=1,N
71 Z(12,L)=Z(11,L)

71 CONTINUE.

72 DO 73 L=1,N
73 Z(11,L)=X

72 CONTINUE.

73 CONTINUE.

74 DO 75 L=1,N
75 X=Z(11,L)

75 CONTINUE.

76 DO 77 L=1,N
77 Z(12,L)=Z(11,L)

77 CONTINUE.

78 DO 79 L=1,N
79 Z(11,L)=X

78 CONTINUE.

79 CONTINUE.

80 DO 81 L=1,N
81 Z(11,L)=Z(11,L)

81 CONTINUE.

82 DO 83 L=1,N
83 Z(12,L)=Z(12,L)

83 CONTINUE.

84 DO 85 L=1,N
85 Z(11,L)=X

84 CONTINUE.

85 CONTINUE.

86 DO 87 L=1,N
87 Z(11,L)=Z(11,L)

87 CONTINUE.

88 DO 89 L=1,N
89 Z(12,L)=Z(12,L)

89 CONTINUE.

90 DO 91 L=1,N
91 Z(11,L)=X

90 CONTINUE.

91 CONTINUE.

92 DO 93 L=1,N
93 Z(11,L)=Z(11,L)

93 CONTINUE.

94 DO 95 L=1,N
95 Z(12,L)=Z(12,L)

95 CONTINUE.

96 DO 97 L=1,N
97 Z(11,L)=X

96 CONTINUE.

97 CONTINUE.

98 DO 99 L=1,N
99 Z(11,L)=Z(11,L)

99 CONTINUE.

100 DO 101 L=1,N
101 Z(12,L)=Z(12,L)

101 CONTINUE.

102 DO 103 L=1,N
103 Z(11,L)=X

102 CONTINUE.

103 CONTINUE.

104 DO 105 L=1,N
105 Z(11,L)=Z(11,L)

105 CONTINUE.

106 DO 107 L=1,N
107 Z(12,L)=Z(12,L)

107 CONTINUE.

108 DO 109 L=1,N
109 Z(11,L)=X

108 CONTINUE.

109 CONTINUE.

110 DO 111 L=1,N
111 Z(11,L)=Z(11,L)

111 CONTINUE.

112 DO 113 L=1,N
113 Z(12,L)=Z(12,L)

113 CONTINUE.

114 DO 115 L=1,N
115 Z(11,L)=X

114 CONTINUE.

115 CONTINUE.
31* 73  \( Z(11,L) \times Z(-11,L) / X \)
32* DO 76 J=1,N
33* IF (I-11) 74,76,74
34* \( Z(J,11) = 0 \)
35* DO 75 L=1,N
36* 75 \( Z(J,L) = Z(J,L) - Z(11,L) \times X \)
37* 76 CONTINUE
38* 74 CONTINUE
39* L = N+1
40* DO 79 I=1,N
41* L = L-1
42* IF (I \times L,11 \times L + 2) 77,79,77
43* 77 I2 = I \times (L,1)
44* I1 = I \times (L,2)
45* DO 78 K=1,N
46* X = Z(K,12)
47* Z(K,12) = Z(K,11)
48* 78 CONTINUE
49* 79 CONTINUE
50* 399 RETURN
51* END

END OF COMPILATION: NO DIAGNOSTICS.
SUBROUTINE STATE(X,D)
COMMON YO,DYO,DB,T,H,NV,NUM
DOUBLE PRECISION YO(10),DY0(10),DB(10)
DOUBLE PRECISION X(1),D(1)

D(2)=X(1)-X(2)
D(1)=1000*X(1)+D(2)
NUM=NUM+1
RETURN
END

END OF COMPILATION: NO DIAGNOSTICS.
<table>
<thead>
<tr>
<th>H</th>
<th>N5</th>
<th>N6</th>
<th>N4</th>
<th>N3</th>
<th>N2</th>
<th>N1</th>
<th>N0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100000E-05</td>
<td>N5</td>
<td>26</td>
<td>N6</td>
<td>4</td>
<td>N4</td>
<td>4</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>5</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>4</td>
<td>N4</td>
<td>4</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>5</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>4</td>
<td>N4</td>
<td>4</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>5</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>6</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>7</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>8</td>
<td>N3</td>
</tr>
<tr>
<td>0.100000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>9</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>10</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>11</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>12</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>13</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>14</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>10</td>
<td>N4</td>
<td>15</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>11</td>
<td>N4</td>
<td>16</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>12</td>
<td>N4</td>
<td>17</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>13</td>
<td>N4</td>
<td>18</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>14</td>
<td>N4</td>
<td>19</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>20</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>21</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>22</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>23</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>24</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>25</td>
<td>N3</td>
</tr>
<tr>
<td>0.200000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>10</td>
<td>N4</td>
<td>26</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>27</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>28</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>29</td>
<td>N3</td>
</tr>
<tr>
<td>0.400000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>30</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>31</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>32</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>33</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>34</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>35</td>
<td>N3</td>
</tr>
<tr>
<td>0.800000E-03</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>36</td>
<td>N3</td>
</tr>
<tr>
<td>0.160000E-02</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>37</td>
<td>N3</td>
</tr>
<tr>
<td>0.160000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>38</td>
<td>N3</td>
</tr>
<tr>
<td>0.160000E-02</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>39</td>
<td>N3</td>
</tr>
<tr>
<td>0.160000E-02</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>40</td>
<td>N3</td>
</tr>
<tr>
<td>0.320000E-02</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>41</td>
<td>N3</td>
</tr>
<tr>
<td>0.320000E-02</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>42</td>
<td>N3</td>
</tr>
<tr>
<td>0.320000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>43</td>
<td>N3</td>
</tr>
<tr>
<td>0.320000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>44</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>45</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>46</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>47</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>48</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>9</td>
<td>N4</td>
<td>49</td>
<td>N3</td>
</tr>
<tr>
<td>0.640000E-02</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>10</td>
<td>N4</td>
<td>50</td>
<td>N3</td>
</tr>
<tr>
<td>0.128000E-01</td>
<td>N5</td>
<td>3</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>51</td>
<td>N3</td>
</tr>
<tr>
<td>0.128000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>52</td>
<td>N3</td>
</tr>
<tr>
<td>0.128000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>53</td>
<td>N3</td>
</tr>
<tr>
<td>0.128000E-01</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>54</td>
<td>N3</td>
</tr>
<tr>
<td>0.256000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>5</td>
<td>N4</td>
<td>55</td>
<td>N3</td>
</tr>
<tr>
<td>0.256000E-01</td>
<td>N5</td>
<td>1</td>
<td>N6</td>
<td>6</td>
<td>N4</td>
<td>56</td>
<td>N3</td>
</tr>
<tr>
<td>0.256000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>7</td>
<td>N4</td>
<td>57</td>
<td>N3</td>
</tr>
<tr>
<td>0.256000E-01</td>
<td>N5</td>
<td>2</td>
<td>N6</td>
<td>8</td>
<td>N4</td>
<td>58</td>
<td>N3</td>
</tr>
<tr>
<td>T</td>
<td>0.2500000E-01 V5 =</td>
<td>2 N6 = 9 N4 = 59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
<td>-----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>2 N6 = 10 N4 = 60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>1 N6 = 11 N4 = 61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>2 N6 = 5 N4 = 62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>2 N6 = 6 N4 = 63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>2 N6 = 7 N4 = 64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.51200E-01 V5 =</td>
<td>2 N6 = 8 N4 = 65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.500000E-01 N5 =</td>
<td>0.779044</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.51200E-01 N5 =</td>
<td>2 N6 = 9 N4 = 66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.51200E-01 N5 =</td>
<td>3 N6 = 5 N4 = 68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.51200E-01 N5 =</td>
<td>1 N6 = 6 N4 = 69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.51200E-01 N5 =</td>
<td>1 N6 = 7 N4 = 70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.51200E-01 N5 =</td>
<td>2 N6 = 8 N4 = 71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.1000000E-01 N5 =</td>
<td>0.606760</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>3 N6 = 5 N4 = 72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 6 N4 = 73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 7 N4 = 74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.1500000E-01 N5 =</td>
<td>0.472373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 8 N4 = 75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>3 N6 = 9 N4 = 76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.2000000E-01 N5 =</td>
<td>0.366961</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 10 N4 = 77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 11 N4 = 78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H = 0.20480E-00 N5 =</td>
<td>2 N6 = 12 N4 = 79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| T = 0.2500000E-01 0.143333 |
| H = 0.40960E 0.286662 |
| T = 0.3000000E-01 0.111111 |
| H = 0.40960E 0.223256 |
| T = 0.3500000E-01 0.124000 |
| H = 0.40960E 0.677053 |
| T = 0.4000000E-01 0.135392 |
| H = 0.40960E 0.869545 |
| T = 0.4500000E-01 0.105446 |
| H = 0.40960E 0.204800 |
| T = 0.5000000E-01 0.821201 |
| H = 0.40960E 0.236346 |
| T = 0.5500000E-01 0.498066 |
| H = 0.40960E 0.249095 |
\[ H = 0.81920E \, 00 \, N_5 = 5 \, N_6 = 5 \, N_4 = 89 \]

\[ T = 0.65000000E \, 01 \]
\[ 0.194006E-01 \, 0.387900E-01 \]

\[ T = 0.70000000E \, 01 \]
\[ 0.151000E-01 \, 0.301892E-01 \]

\[ H = 0.81920E \, 00 \, N_5 = 4 \, N_6 = 6 \, N_4 = 90 \]

\[ T = 0.75000000E \, 01 \]
\[ 0.117309E-01 \, 0.234568E-01 \]

\[ H = 0.81920E \, 00 \, N_5 = 2 \, N_6 = 7 \, N_4 = 91 \]

\[ T = 0.80000000E \, 01 \]
\[ 0.911613E-02 \, 0.182281E-01 \]

\[ T = 0.85000000E \, 01 \]
\[ 0.785756E-02 \, 0.141683E-01 \]

\[ H = 0.81920E \, 00 \, N_5 = 2 \, N_6 = 8 \, N_4 = 92 \]

\[ T = 0.90000000E \, 01 \]
\[ 0.550849E-02 \, 0.110139E-01 \]

\[ T = 0.95000000E \, 01 \]
\[ 0.428615E-02 \, 0.856983E-02 \]

\[ H = 0.81920E \, 00 \, N_5 = 2 \, N_6 = 9 \, N_4 = 93 \]

\[ T = 0.10000000E \, 02 \]
\[ 0.333520E-02 \, 0.666902E-02 \]

STATE EQUATIONS EVALUATED 278 TIMES

JACOBIAN EVALUATED 4 TIMES