A STUDY OF INTEGRATION ALGORITHMS IN CHEMICAL AND PHYSIOLOGICAL SYSTEM DYNAMICS

A Thesis

Presented to

the Faculty of the Department of Chemical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Francisco Sahagún Castellanos

December 1974

to my mother and my father,

to my lovely wife, María,

and to my children

ACKNOWLEDGEMENTS

The author wishes to express sincere gratitude to Dr. R. L. Motard for his guidance in the preparation of this work.

The suggestions, comments and help of all kind provided by Dr. M. Schacham and Akhil Bidani are very truly appreciated.

I wish to thank the Chemical Engineering Department for the education provided, and both the Chemical Engineering Department and IBM de Mexico S. A. for the financial assistance provided.

I am especially thankful to Ms. Sandi White for her patience and help in the preparation of this thesis.

I would like also to express my gratitude to Professor Armando García, David K. Lowell and Donald C. Clark for their help at various stages of this work.

A STUDY OF INTEGRATION ALGORITHMS IN CHEMICAL AND PHYSIOLOGICAL SYSTEM DYNAMICS

An Abstract of a Thesis

Presented to

the Faculty of the Department of Chemical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Francisco Sahagún Castellanos

December 1974

ABSTRACT

Solution of many chemical engineering problems requires the use of numerical integration techniques. The most popular integration technique for such problems is the classical fourth order Runge-Kutta that in some cases can be used only with very low efficiency.

In the last few years new methods have been developed which seem to be efficient for regular and stiff problems. Some of these new methods were compared in the solution of chemical and physiological systems. The program written by C. W. Gear, which includes two slightly different algorithms for stiff systems and a third algorithm for regular systems was found to be the most stable and efficient in all cases.

To increase accessibility of the Gear program for general engineering usage, a set of subroutines was written. These subroutines were designed with the following objectives:

a) Minimization of input requiremtns.

b) Minimization of interaction between user and integration program.

c) Giving the user the option of calculating the values of the dependent variables at certain specific values of the independent variable.

d) Giving the user the option of dynamically selecting the algorithm to solve the problem in question efficiently.

The behavior of the different algorithms in the solution of selected problems is also discussed.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Solution of many chemical engineering problems requires the use of numerical integration techniques. The most popular integration technique for such problems is the classical fourth order Runge-Kutta. The main advantage of this method is that it can be easily programmed and its efficiency is reasonable for simple problems. However in many chemical engineering problems this method cannot be used at all, or may only be used with very low efficiency.

Problems like simulation of a tubular reactor [3] or simulation of batch distillation columns [6] are stiff problems which have very large negative eigenvalues. For such problems the Runge-Kutta method is less efficient than the simple Euler method. Also for regular (non-stiff) problems, some of the predictor corrector methods proved to be less time consuming than the Runge-Kutta method.

In the last few years new methods have been developed [4] which seem to be efficient for regular and stiff problems.

The aim of this work is:

1) To compare the efficiency of the new methods for typical chemical engineering problems.

2) To prepare a general purpose integration subroutine which will be easy to use and will solve chemical engineering problems efficiently.

Chapter 1

NUMERICAL INTEGRATION GENERAL CONCEPTS

The natural laws in any scientific or technological field are not regarded as precise and definitive until they have been expressed in mathematical form. Such a form, usually an equation, is a relation between the quantity of interest, say product yield, and independent variables such as time and temperature upon which yield depends. When it occurs that this equation involves, in addition to the function itself, one or more of its derivatives it is called a differential equation. Except in those few cases in which the differential equations can be integrated directly, the solution will have to be approximated using numerical integration techniques.

An approach to the numerical approximation for the solution of differential equations is the difference method. The solution is approximated by its value at a sequence of discrete points called the mesh points. Ideally the solution could be represented by its actual value at each mesh point so that it could be approximated to high accuracy by interpolation between the mesh points. However, two problems interfere with this ideal:

1) The exact solution of the differential equation is not in general known and cannot be calculated, so the

solution to a different problem which can be calculated is sought. Thus the truncation error can be defined as the error introduced by using a mathematical formula which is an approximation to the true problem and is a direct function of the interval size, the numerical method and the particular system under consideration.

2) Numbers cannot be represented exactly in the numerical processes involved. The change introduced by this mechanism will be called round-off error. Rounding of numbers is a necessary evil in machine computation because of the fixed digit capacity of the memory and arithmetic units. We have to consider also the local truncation error defined as the amount by which the solution of the differential equation fails to satisfy the equation used in the numerical method.

Consequently the difference method solution will be represented by a finite precision number containing two sources of error, round-off and truncation.

When we think of approximating a solution numerically we naturally are concerned with how accurate we can make the numerical solution to the actual solution. When we pick a method it may depend on one or more parameters, for example, the step size or the number of terms in a series expansion. We would like to know how to pick these parameters to achieve any desired accuracy. It is possible that there is an error below which it is not possible to go.

Before attempting to numerically integrate a system of differential equations, we have to be sure that the system has a unique solution. This can be done using the following theorem: (A proof of which can be found in [12].)

If $y' = f(y,t)$ is a differential equation such that $f(y,t)$ is continuous in the region $0 \leq t \leq b$, and if there exists a constant L such that

$$|f(y,t) - f(y^*,t)| \leq L|y - y^*| \tag{1}$$

where (y,t) and (y^*,t) are two different points in the region $0 \leq t \leq b$ with the same value for t for all $0 \leq t \leq b$ and all y, y^*.

Then there exists a unique continuously differentiable function $y(t)$ such that

$$y'(t) = f\big(y(t),t\big)$$
$$\text{and} \quad y(0) = y_0, \text{ the initial condition.} \tag{2}$$

The condition expressed by equation (1) is called the Lipschitz condition and L is called the Lipschitz constant.

In addition to insure that the problem has a solution, we have also to assure ourselves that it is "well posed". By this we mean that small perturbations in the stated problem will only lead to small changes in the answers. Fortunately it can be shown that the Lipschitz condition for an ordinary initial value problem is a sufficient condition to be "well posed".

Any desired degree of accuracy can be achieved for any problem satisfying a Lipschitz condition by picking a small enough step size. Since as the step size decreases the number of points and hence the amount of calculations increases, we would expect the effect of round-off errors to increase because there are more of them. Therefore in defining convergence, we must require that the computations indicated in the method be performed exactly. In practice, this means that additional digits are carried in the computations as the step size decreases.

At each step of the numerical integration, the process is considered to have converged when each component of the dependent variable vector has satisfied a convergence criterion. In testing for convergence, two successive values (for each component) at the point in question are compared. Let the difference between the two for the "jth" component be δ_j, and let the error tolerance allowed by the user be EP. Three convergence error criteria can then be stated as:

1) Standard error

Let $(y_j)_M$ be the largest absolute value attained so far in the integration by the dependent variable y_j. The convergence requirement is:

$$|\delta_j/(y_j)_M| < EP \qquad j = 1,2,\ldots,n \tag{3}$$

if $(y_j)_M < EP$, it is replaced by EP.

2) Relative error

Let y_j be the current approximation to the respective dependent variable. The convergence requirement is:

$$|\delta_j/y_j| < EP \qquad j = 1,2,\ldots,n \tag{4}$$

if $|y_j| < EP$, it is replaced by EP in the test.

3) Absolute error

The convergence requirement is:

$$|\delta_j| < EP \qquad j = 1,2,\ldots,n \tag{5}$$

The concept of stability is associated with the propagation of errors of the numerical technique as the calculations progress with a finite interval size, that is, the effect errors made on one step will have on suceeding steps. Convergence does not necessarily imply stability.

The problem of instability arises because in most instances the order of the approximation difference equation is higher than that of the original differential equation. Hence, the difference equation possesses extraneous solutions which in some instances can dominate the solution, so that the solution of the difference equation bears little if any resemblance to the true solution of the original differential equation. It happens frequently that the spurious solutions do not vanish even in the limits as the increment sizes approach zero. This phenomenon is called strong instability, and implies lack of convergence as well as lack of stability. When a method possesses convergence but has unstable asymptotic behavior, the phenomenon is called weak instability. If the extraneous solutions tend to grow as the calculations progress, but at a rate much slower than the true solution, the solution is said to posess relative stability; if the extraneous solutions tend to grow as the calculation progresses, at a rate faster than the true solution, the solution is said to be unstable.

The methods which require a knowledge of the differential equations and initial values only are called one-value methods. They are usually called one-step methods because they only require the value at one mesh point to compute the value at the next.

There are other kinds of methods which require several pieces of information about the dependent variable at time $t = t_{n-1}$ in order to compute the equivalent pieces of information at $t = t_n$. We call these methods multi-value methods since they use more than one value of the dependent variable. Often these methods use the values of dependent variable and its derivatives at "k" different mesh points $t_{n-1}, t_{n-2}, \ldots, t_{n-k}$. They are therefore called multi-step methods.

A multi-value method consists of two processes which are called prediction and correction. In the predictor process an approximation to $\bar{y}_n$ is computed from $\bar{y}_{n-1}$ by linear extrapolation. This approximation is called $\bar{y}_{n,(0)}$ and it is given by

$$\bar{y}_{n,(0)} = \bar{\bar{B}}\, \bar{y}_{n-1} \tag{6}$$

where $\bar{\bar{B}}$ is any suitable matrix of constants.

The prediction process does not make use of the differential equation in any way, so the correction process corrects the approximate values if they do not satisfy the differential equation at $t = t_n$. The differential equation is written as:

$$0 = G(\bar{y}_n) = -(\bar{y}_n)_k + hf\left((\bar{y}_n)_0\right) = -hy_n' + hf(y_n) \qquad (7)$$

where $(\bar{y})_i$ is the ith component of the vector $\bar{y}$. $G(\bar{y}_{n,n(0)})$ is the amount by which $\bar{y}_{n,n(0)}$ does not satisfy the differential equation.

A vector multiple of this scalar is added to $\bar{y}_{n,(0)}$ to correct it by the process

$$\bar{y}_{n,(1)} = \bar{y}_{n,(0)} + \bar{c}G(y_{n,(0)}). \qquad (8)$$

This process can be repeated by

$$\bar{y}_{n,(m+1)} = \bar{y}_{n,(m)} + \bar{c}G(\bar{y}_{n,(m)}) \quad m = 1,2,\ldots,M \qquad (9)$$

for a fixed number of iterations or until there is no further change in $\bar{y}_{n,(m)}$. The value used for $\bar{y}_n$ is then $y_{n,(M)}$, where M is either fixed or large enough to get convergence, that is such that $G(y_{n,(M)})$ is zero to the accuracy desired. We say that M is the number of corrector iterations.

We can distinguish two special cases of multi-value methods:

1) Explicit

When the method provides an explicit way of computing y_n and hy_n' from the values of y and its derivatives at preceding points. It requires only a vector inner product and one evaluation of the function f for each step.

2) Implicit

When the method is represented through a non-linear equation involving the function f and that must be solved for y_n, we have an implicit method.

In accordance to the magnitudes of the eigenvalues, we have regular systems and stiff systems of differential equations. Stiff systems are those in which the magnitudes of the eigenvalues vary greatly, and when this happens the system has components whose values will be quickly very small if compared with the values of other components. To guarantee a good solution most of the numerical integration methods require that the absolute value of the product of the interval size and the eigenvalue be bounded by a single small number, the order of which is from 1 to 10. As a result, an extremely small interval size has to be used over the entire range of integration and the computation time can become excessive.

1.1 Stiff Stability

Gear [1] defined stiff stability as follows:

A method is stiffly stable if in the region $R_1 (Re(h\lambda) \leq D)$ it is absolutely stable, and in $R_2 (D < Re(h\lambda) < \alpha, |Im(h\lambda)| < \theta)$ it is accurate.

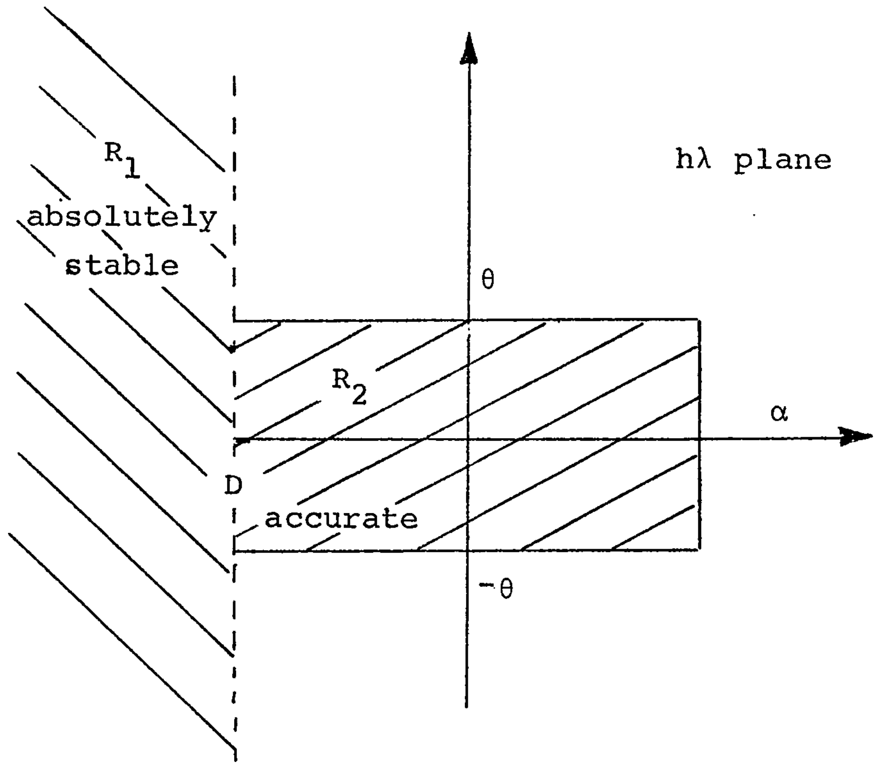


Figure 1.1
Stiff Stability

The rationale for this definition is as follows: $e^{\lambda h}$ is the change in a component in one step due to an eigenvalue λ. If $h\lambda = u + iv$, then the change in magnitude is e^u. If $u < D < 0$, then the component is reduced by at least e^D in one step. We are not interested in the accuracy of components that are very small, so for some D we are willing to ignore all components in R_1. We just require that the method be absolutely stable. Around the origin we are interested in accuracy, for which relative or absolute stability is necessary.

1.2 Systems of Equations and Equations of Order Greater than One

One common technique for handling equations of the form:

$$y^{(p)} = f(y,y',\ldots,y^{(p-1)},t) \tag{10}$$

where f satisfies a Lipschitz condition in each of the $y^{(i)}$, and $y^{(i)}$ is a notation for d^iy/dt^i, is to transform them into an equivalent first order system. If we define the variables:

$$y^i = y^{(i-1)} \qquad i = 1,2,\ldots,p \tag{11}$$

we can write (10) as

$$(y^p)' = f(y^1,y^2,y^3,\ldots,y^p,t). \tag{12a}$$

While by differentiation of (11) for i = 1,2,...,p-1 we get:

$$(y^i)' = y^{(i)} = y^{i+1} \tag{12b}$$

Equation (12) is a system of p first order equations which can be handled by the methods used to solve first order equations.

The original problem (equation 10) will have initial values specified for $y^{(i-1)}(0)$, $i = 1,2,\ldots,p$ so (12) will have initial values specified for $y'(0)$ as required.

Chapter 2

SELECTION AND DESCRIPTION OF METHODS

2.1 Selection of Methods

Among the recommendations made by Hull et al. [4], the following were considered.

If a program library is to provide the best available method for whatever situations may arise, it must contain several different methods, including:

1) An extrapolation method for situations in which function evaluations are relatively inexpensive.

2) A variable order Adams method for situations in which function evaluations are expensive.

3) A fourth or fifth order Runge-Kutta method for calculations in which the function evaluations are simple and the tolerance not very stringent.

4) The library should also contain special methods for special problems. The most important of these is a special method for stiff systems.

5) It also may be helpful to have special methods that can be applied directly to higher order differential equations.

Regarding extrapolation methods, only the one originated by Bulirsch and Stoer was tested and it is therefore the only one that can be recommended.

The variable order Adams method originated by Gear can also be recommended because it is almost as efficient as the Krogh's, but in addition to that it includes an option for handling stiff systems that is very efficient.

In accordance with these recommendations the following methods were chosen:

The extrapolation methods selected are the implementation presented by Gear [2] of the Bulirsch and Stoer rational function extrapolation which also includes the polynomial extrapolation method and the Bulirsch and Stoer algorithm modified by Crane and Fox [7].

The Gear implementation of the variable order Adams method was selected because it is very efficient and contains in the same program the option for stiff systems.

One of the most important attractions of using Runge-Kutta methods is that they are quite a bit easier to program than variable order Adams methods. The fourth order Runge-Kutta was selected because Runge-Kutta of order higher than fourth requires a more complicated algebraic work.

2.2 Description of Methods

Fourth Order Runge-Kutta

The Runge-Kutta of the fourth order approximates the differential equation with a Taylor's series expansion of order 4 but it is derived in such a way that all higher order terms are expressed in terms of first order derivatives.

This method employs a recurrence formula as follows:

$$y_{i+1} = y_i + a_1k_1 + a_2k_2 + a_3k_3 + a_4k_4 \tag{13}$$

to calculate successive values of the dependent variable of the differential equation

$$y' = f(x,y) \tag{14}$$

where $k_1 = hf(x_i,y_i)$

$$\begin{aligned} k_2 &= hf(x_i + p_1h, y_i + q_{11}k_1) \\ k_3 &= hf(x_i + p_2h, y_i + q_{21}k_1 + q_{22}k_2) \\ k_4 &= hf(x_i + p_3h, y_i + q_{31}k_1 + q_{32}k_2 + q_{33}k_3). \end{aligned} \tag{15}$$

The p's and q's assume values such that (13) becomes

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{16}$$

The principal advantage of the Runge-Kutta methods is the self-starting feature and resulting ease of programming, and one disadvantage is the requirement that the function f(x,y) must be evaluated for several slightly different values of the independent variable in every step of the solution. This repeated determination of f(x,y) usually results in a less efficient method with respect to computing time, than do other methods of comparable accuracy, in which previously determined values of the dependent variable are used in subsequent steps. Another disadvantage is that it is more difficult to estimate the per-step error for higher order Runge-Kutta solutions than for solutions obtained by some other commonly used methods.

The computer program uses the Runge-Kutta-Merson method which is a fourth order Runge-Kutta process that simultaneously gives an approximation to the single step error.

Merson [8] has suggested the following equations:

$$\eta_0 = y_n \qquad k_0 = hf(\eta_0)$$

$$\eta_1 = \eta_0 + \frac{k_0}{3} \qquad k_1 = hf(\eta_1)$$

$$\eta_2 = \eta_0 + \frac{k_0 + k_1}{6} \qquad k_2 = hf(\eta_2)$$

$$\eta_3 = \eta_0 + \frac{k_0 + 3k_2}{8} \qquad k_3 = hf(\eta_3) \qquad (17)$$

$$\eta_4 = \eta_0 + \frac{k_0 - 3k_2 + 4k_3}{2} \qquad k_4 = hf(\eta_4)$$

$$y_{n+1} = \eta_5 = \eta_0 + \frac{k_0 + 4k_3 + k_4}{6}$$

or

$$y_{n+1} = y_n + \frac{k_0 + 4k_3 + k_4}{6}$$

The additional computation serves to determine the error, in an approximate sense. If the interval is small enough, so that we can represent f(x,y) by the linear approximation:

$$f(x,y) = Ax + By + c. \qquad (18)$$

Merson shows that

$$\eta_4 = y(t_{n+1}) - \frac{1}{120}h^5 y^{(5)} + 0(h^6) \tag{19}$$

and
$$\eta_5 = y(t_{n+1}) - \frac{1}{720}h^5 y^{(5)} + 0(h^6). \tag{20}$$

Thus the difference $\eta_5 - \eta_4$ is an indication of the local error. This method has been used successfully in a number of automatic step selection methods.

Bulirsch and Stoer and Polynomial Extrapolation (DIFSUB)

Two algorithms are used in this program:

1) Rational function extrapolation
2) Polynomial extrapolation

The original rational extrapolation algorithm was proposed by Bulirsch and Stoer [9]. The particular implementation we have used is based in the FORTRAN version by Clark [10] of the Bulirsch and Stoer ALGOL program. This algorithm was certified and modified slightly by Crane and Fox [7], and such a version corresponds to the program DESUB that has also been used in this work.

A rational function is one which is the quotient of two polynomials. The algorithm uses diagonal rational polynomials (i.e., rational functions of the form:

$$R_m(h) = \frac{P_m(h)}{Q_m(h)} = \frac{P_0 + P_1 h + \ldots + P_u h^u}{q_0 + q_1 h + \ldots + q_v h^v}$$

where u = integer part of m/2

v = integer part of (m+1)/2).

Defining $R_m^i(h)$ as the rational approximation which agrees with $y(x,h)$ at $h = h_i, h_{i+1}, \ldots, h_{i+m}$, where $h_i > h_{i+1} > \ldots > h_{i+m}$ and $R_m^i(0) = R_m^i$, then the R's can be obtained by the formula:

$$R_{-1}^i = 0 \text{ (This is used to get the recurrence started.)}$$

$$R_0^i = y(t,h_i)$$

$$R_m^i = R_{m-1}^{i+1} + \frac{R_{m-1}^{i+1} - R_{m-1}^i}{\left(\frac{h_i}{h_{i+m}}\right)^2 \left[1 - \frac{R_{m-1}^{i+1} - R_{m-1}^i}{R_{m-1}^{i+1} - R_{m-2}^{i+1}}\right] - 1} \qquad m \geq 1 \tag{21}$$

This formula involves calculating the differences of numbers that are getting closer and closer to the answer R_∞^0, so less floating point error is obtained by calculating the differences directly.

Thus (21) can also be written:

$$D_m^i = \frac{C_{m-1}^{i+1} \cdot W_{m-1}^{i+1}}{\left(\frac{h_i}{h_{i+m}}\right)^2 D_{m-1}^i - C_{m-1}^{i+1}} \qquad m \geq 1 \tag{22}$$

$$C_m^i = \frac{\left(\frac{h_i}{h_{i+m}}\right)^2 D_{m-1}^i \cdot W_{m-1}^{i+1}}{\left(\frac{h_i}{h_{i+m}}\right)^2 D_{m-1}^i - C_{m-1}^{i+1}} \qquad m \geq 1 \tag{23}$$

and $$W_m^i = C_m^i - D_m^{i-1} \tag{24}$$

where $D_m^i = R_m^i - R_{m-1}^{i+1}$ (25)

$$C_m^i = R_m^i - R_{m-1}^i \qquad (26)$$

$$W_m^i = R_m^i - R_m^{i-1}. \qquad (27)$$

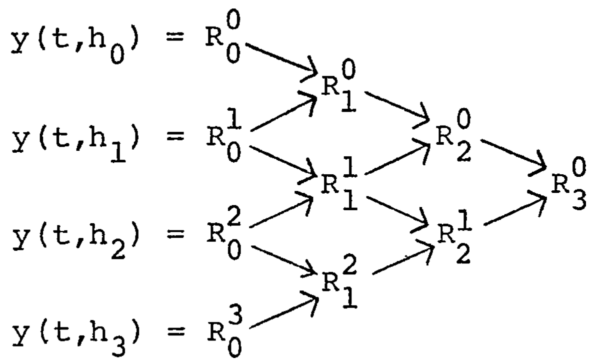
Polynomial extrapolation, the other algorithm used in DIFSUB is performed by integrating the differential equation over the interval (0,t) using step size h_i to get the results y(t,h) for i = 0,1,2,...,m where $h_0 > h_1 > h_2 > \ldots > h_m > 0$. The polynomial in "h" of "m" that passes through these results is then evaluated at h = 0.

The Aitken interpolation process allows a rule for constructing a triangle of approximations to $R_m^i(h)$ in which two values are used to form the next approximation by the relation:

$$R_m^i = R_{m-1}^{i+1} + \frac{R_{m-1}^{i+1} - R_{m-1}^i}{\left(\dfrac{h_i}{h_{i+m}}\right) - 1} \qquad (28)$$

where $R_m(t,h)$ is a polynomial of degree m in h.

Aitken Interpolation



As can be seen this type of method has the potential for being of arbitrarily large order (by making "m" larger) and for reducing the step without voiding the results of work at larger steps (by increasing "i" for fixed "m"). In practice, these procedures are limited by rounding errors and instabilities. Increasing "m" or "i" increases the number of calculations in the basic integration by the Euler method at the rate of 2^{m+1}. This means that precision will be lost as "i" or "m" increase.

The stability of the polynomial extrapolation methods is still an open question. Experience has shown that equations with very negative values of $\partial f/\partial y$ cause severe instability problems until the basic step size "h_0" is reduced to the point where

$$\left|h_0 \frac{\partial f}{\partial y}\right| \text{ is close to 1.}$$

Hull [4] says that the Bulirsch and Stoer method is particularly good when the functions evaluations are not very expensive.

Crane and Fox "DESUB"

The algorithm used in this program to integrate a set of first order ordinary differential equations is based on an extrapolation procedure. Extrapolation to zero interval size is a familiar device for improving a discrete approximation, T(h), whose truncation error can be expressed in powers of the discretization interval "h".

Let

$$T(h) = t_0 + t_1h^{p_1} + t_2h^{p_2} + \ldots. \tag{29}$$

where t_0 is the exact answer and the rest of the sum represents the truncation error. In extrapolation, a sequence of h's, $h_0, h_1, h_2, \ldots.$ tending to zero is used to compute successive approximations $T(h_0), T(h_1), T(h_2), \ldots.$. The first two approximations can be used to eliminate the term of order h^{p_1}, the next one to eliminate the term of order h^{p_2}, etc. In other words, a polynomial of powers in "h" is fitted to the approximations and extrapolated to zero interval size, $h = 0$.

Instead of a polynomial a rational function of "h" may be used and extrapolated to zero, often with improved convergence. The program DESUB is based on the use of rational functions as developed by Bulirsch and Stoer [9]. Rational extrapolation is used in their algorithm, to a modified midpoint integration rule.

Let

$$\frac{dy_i}{dx} = f(x, y_1, y_2, \ldots, y_n) \qquad i = 1,2,\ldots,n \tag{30}$$

be the system of differential equations to be integrated with the initial conditions

$$y_i(0) = y_{i0} \qquad i = 1,2,\ldots,n. \tag{31}$$

The algorithm described below for a single equation

$$\frac{dy}{dx} = f(x,y) \tag{32}$$

applies, for a system, to each equation in the system.

The modified midpoint rule at any point is based on a "lower" value, "y_l" of "y", and a middle value, "y_m", of "y" at an interval "h/2" ahead of "y_l". These values are used to compute an "upper value", "y_u", of "y" at an interval "h/2" ahead of "y_m". The value "y_u" is computed from

$$y_u = y_l + h \cdot f(x_m, y_m). \tag{33}$$

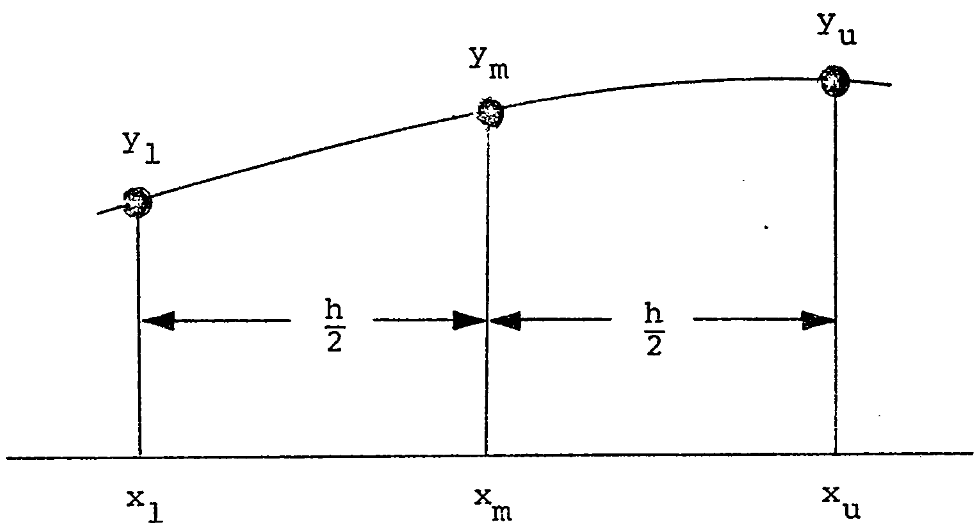


Figure 2.1 Modified Midpoint Rule

At the next step "y_m" becomes "y_l", "y_u" becomes "y_m", and the process continues until the final "y_m" is the value at the end of the integration interval. A final correction sets the value of "y" at this point to an average.

$$y = \frac{1}{2}\left[y_m + \left(y_l + \left(\frac{h}{2}\right) f(x_m, y_m)\right)\right] \tag{34}$$

To start the process the algorithm uses:

$$y_1 = y_0$$
$$y_m = y_0 + \left(\frac{h}{2}\right) f(x_0, y_0). \tag{35}$$

Under suitable differentiability conditions the modified midpoint approximation to the solution y(x) has the asymptotic expansion:

$$T(h,x) = y(x) + t_1(x)h^2 + t_2(x)h^4 + \ldots. \tag{36}$$

and an extrapolation to zero interval size can be applied. A rational extrapolation of order 6 is used in the programmed implementation; that is at each integration step as many as six applications of the midpoint rule are computed for sucessively smaller "h" and extrapolated to h = 0 in attempting to achieve convergence. This method allows the user to make his own choice among three different convergence criteria.

GEARSB

This program includes two methods:

1) Adams predictor-corrector

2) A multi-step method suitable for stiff systems

The Adams predictor-corrector formulas are as follows:

A) K-step Adams-Bashfort formula (explicit multistep method):

$$y_n = y_{n-1} + h \sum_{j=0}^{k-1} \gamma_j \nabla^j f_{n-1} \tag{37}$$

which expresses an approximation to the value of y(t) at t_n in terms of the value at t_{n-1} and the backwards differences of the derivatives at "t_{n-1}".

The backwards differences are given by:

$$\nabla^{q+1} f_m = \nabla^q f_m - \nabla^q f_{m-1} \tag{38}$$

where $\nabla^0 f_m = f_m$

$$\gamma_j = (-1)^j \int_0^1 \binom{-s}{j} ds.$$

The Adams-Bashfort formula can also be written as:

$$y_n = y_{n-1} + h \sum_{i=1}^{k} \beta_{ki} f_{n-1} \tag{39}$$

where $\beta_{ki} = (-1)^{i-1} \sum_{j=i-1}^{k-1} \gamma_j \binom{j}{i-1}.$

B) Adams-Moulton method formula (implicit multi-step method)

$$y_n = y_{n-1} + h \sum_{j=0}^{k-1} \gamma_j^* \nabla^j f_n \tag{40}$$

where $\gamma_j^* = (-1)^j \int_0^1 \binom{-s+1}{j} ds.$

This formula can also be expressed in the following form:

$$y_n = y_{n-1} + h \sum_{i=0}^{k-1} \beta_{ki}^* f_{n-i} \tag{41}$$

where $\beta_{ki}^* = (-1)^j \sum_{j=1}^{k-1} \binom{j}{i} \gamma_j^*.$

The region of stability for the implicit Adams-Moulton methods is larger by a factor of ten or more than that of the explicit Adams-Bashfort method. The truncation errors are also smaller for the implicit methods, so the implicit methods can be used with a step size that is several times larger than that of the explicit methods.

The local truncation error is proportional to a derivative of the solution "y(t)". The derivative can be estimated by using a numerical differentiation formula.

If it is desired to control the single step truncation error to be less than ε, we must select h so that:

$$C_{q+1}\, q!\, \nabla a_q \leq \varepsilon \tag{42}$$

where ∇a_q is the backwards difference of the last component of a and $a = [y, hy', \ldots, h^q y^{(q)}/q!]^T$.

When a system of equations is to be integrated, it may be desirable to control the error in each component differently, so we control

$$C_{q+1}\, q!\, \left|\left|\frac{\nabla a_q}{\omega}\right|\right|_2 \leq \varepsilon \tag{43}$$

where for each member of the system there is a ∇a_q component and a weight component ω. $||\cdot||_2$ is the L_2 norm which is defined:

$$||\bar{a}||_2 = \sqrt{\sum_i |a^i|^2}$$

where the a^i are the components of the vector $\bar{a}$.

The basic step control mechanism is to execute one step and to perform the test (43). If the test suceeds, the step is accepted; otherwise it is rejected. The step size to use for the next step or to repeat the rejected step is estimated to be αh where

$$C_{q+1}\, q!\, \alpha^{q+1} \left\| \frac{\nabla a_q}{\omega} \right\|_2 = \varepsilon. \tag{44}$$

If the step size were used and if the error were exactly proportional to h^{q+1} (that is if ∇a_q is constant from step to step), the test would just be satisfied next time. However, ∇a_q is not usually constant, so a slightly smaller step is used in order that test (43) can reasonably be expected to be satisfied. In the program, α is estimated by:

$$\alpha = \frac{1}{1.2} \left[\frac{\varepsilon}{C_{q+1}\, q!} \; \frac{1}{\left\| \frac{\nabla a_q}{\omega} \right\|_2} \right]^{\frac{1}{q+1}} . \tag{45}$$

It is also necessary to test the step size that could be used in other orders.

Since

$$\begin{aligned} \nabla^2 a_q &\cong \frac{h^{q+2}}{q!}\, y^{(q+2)} \\ a_q &\cong \frac{h^q}{q!}\, y^{(q)} , \end{aligned} \tag{46}$$

the step sizes that could be used in orders q+1 and q-1 can be estimated to be αh, where

$$\alpha = \frac{1}{1.4} \left[\frac{\varepsilon}{C_{q+2}\, q!} \; \frac{1}{\left\| \frac{\nabla^2 a_q}{\omega} \right\|_2} \right]^{\frac{1}{q+2}} \quad \text{For order } q+1 \tag{47}$$

$$\alpha = \frac{1}{1.3} \left[\frac{\varepsilon}{C_q\, q!} \; \frac{1}{\left\| \frac{a_q}{\omega} \right\|_2} \right]^{\frac{1}{q}} \quad \text{For order } q-1. \tag{48}$$

The factors 1.3 and 1.4 are to provide a similar range in which test (43) will succeed. They were chosen on an ad hoc basis to bias the method in favor first of not changing the order since it requires additional computer time, and then in favor of reducing the order because it is a little less work per step.

The estimates of α are made:

1) If a step fails, except then no attempt is made to increase the order.

2) q+1 steps after the last change in order or step size.

3) Ten steps after the α were last estimated if no step increase was made at that time. (This is to reduce the overhead of testing too frequently.)

When the α are estimated, the order corresponding to the largest is chosen and the step changed appropriately. The program does not increase the step at the current order q if $\alpha \leq 1.1$ since it is felt that the increase is not worth the computer time to perform it.

Starting is almost automatic. The value of hy_0' can be calculated from the initial value and the differential equations. This is sufficient to allow a first order process to be used. The order control mechanism can then increase the order to a desirable level.

The estimated single step errors are controlled so as to be less than the error test constant (EPS), thus, the overall error is proportional to the number of steps if f is independent of y. If the equations are stable, early errors are decreased so the error is smaller; if they are unstable, the error is larger. This is because the error control is based on estimates of the local truncation error only.

Multistep Method Suitable for Stiff Systems:

In order to express the order and the error coefficient in a convenient way, define the polynomials

$$\rho(\xi) = \sum_{i=0}^{k} \alpha_i \, \xi^{k-1}$$
$$\sigma(\xi) = \sum_{i=0}^{k} \beta_i \, \xi^{k-1}. \tag{49}$$

For a stiffly stable method, $\sigma(\xi)$ must have at least as great a degree as $\rho(\xi)$. Otherwise, one root at $\mu = \infty$ is ∞. This means that the methods are implicit. Consequently we have to solve the corrector equation. For an implicit multi-step method:

$$y_n^{(m+1)} = \sum_{i=1}^{k} (\alpha_i^* \, y_{n-1}^{(m)} + \beta_i^* \, hy'^{(m)}_{n-1} + \beta_0^* \, hf(y_n^{(m)}) \tag{50}$$

which converges if

$$\left|\left| h\beta_0^* \frac{\partial f}{\partial y} \right|\right| < 1. \tag{51}$$

Unfortunately in this case $h(\partial f/\partial y)$ can be very negative, so we must use a different iteration. A Newton solution can be expensive if the evaluation of $\partial f/\partial y$ is expensive, but $\partial f/\partial y$ need not be re-evaluated at each iteration if it does not change much. If a Newton type method in which $\partial f/\partial y$ is not re-evaluated as each step converges, it goes to a solution of the equation. When the method is expressed in a (k+1) value normal form, we usually perform the corrector iteration:

$$\bar{a}_{n(m+1)} + \bar{a}_{n,(m)} + \bar{I}F(a_{n,(m)}). \tag{52}$$

Hence, if the method converged, it would converge to

$$\bar{a}_n = \bar{a}_{n(0)} + \bar{I}\omega \tag{53}$$

where ω is a scalar such that

$$F(\bar{a}_n) = F(\bar{a}_{n(0)} + \bar{I}\omega) = 0. \tag{54}$$

Solving this expression by a Newton iteration, and writing

$$a_{n,(m)} = a_{n,(0)} + I\omega_{(m)}$$

we have

$$\bar{a}_{n(m+1)} = \bar{a}_{n(m)} - \bar{I}\left[\frac{\partial F}{\partial \bar{a}} \cdot \bar{I}\right]^{-1} F(\bar{a}_{n,(m)}) \tag{55}$$

$$\bar{\bar{W}} = \left[\frac{\partial F}{\partial \bar{a}} \cdot I\right] = \left[-L_1 + hL_0 \ \frac{\partial f}{\partial y}\right]^{-1}. \tag{56}$$

$\bar{\bar{W}}$ depends on the order of the method (via β_0), h and $\partial f/\partial y$. If $\partial f/\partial y$ is slowly varying, $\bar{\bar{W}}$ will not change much during the iteration of (55) for a single step or over several steps in which the step and order do not change. The $\bar{I}$ corresponding to $\sigma(\xi) = \xi^k$, $1 \leq k \leq 6$ is selected.

The matrix $\bar{\bar{W}}$ is re-evaluated only if the order is changed or if the corrector fails to converge in the sense that the corrections $\bar{\bar{W}}F(a_{n,(m)}$ are not small by the third iteration.

Hull et al. [4] have worked with this method and have found that it is very efficient for stiff systems.

Chapter 3

STATEMENT AND DISCUSSION OF RESULTS OF THE PROBLEMS SELECTED

Once the programs mentioned in Chapter 2 were modified to fit our computer system and properly implemented, we proceeded to consider several typical chemical engineering problems and we selected the following:

a) A system of reaction rate equations.

b) A stirred tank reactor with exothermic reaction.

c) A turbulent boundary layer on a flat plate.

d) A periodic chemical reaction.

    1. Sinusoidal forcing function

    2. Bang-bang forcing function

e) The thermal decomposition of ozone.

f) The transient behavior of a catalytic fluidized bed.

g) A biological system.

The four computer programs provide the following algorithms, each implementing double precision arithmetic.

1) DIFSUB

    a) Bulirsch and Stoer Rational Extrapolation.

    b) Polynomial Extrapolation.

2) DESUB

    Modified Bulirsch and Stoer Rational Extrapolation.

3) GEARSB

a) Adams Multi-step Predictor Corrector.

b) Gear's higher order implicit method for stiff systems (option 1). For this algorithm, the user must provide a subroutine to evaluate the Jacobian of the system. (For $\bar{x}' = \bar{f}(\bar{x})$, Jacobian $\bar{\bar{J}}(\bar{x}) = \partial\bar{f}/\partial\bar{x}$.)

c) Gear's method for stiff systems (option 2). This algorithm has its own subroutine to compute the partial derivatives by numerical approximation.

4) Runge-Kutta-Merson (fourth order).

DESUB was also implemented in single precision to observe its behavior in both cases.

Considering that the program GEARSB offers two slightly different algorithms for stiff systems (for one of which the user must provide analytically the terms of the Jacobian, while the other algorithm frees the user of this often formidable task by numerically approximating the Jacobian), we decided to use both algorithms and observe the difference in accuracy as well as in computation time caused by this factor. Thus, we tried to solve each problem with eight algorithms, using error allowances of 1×10^{-3}, 1×10^{-5} and 1×10^{-7}. In some cases these error allowances had to be automatically increased in order to avoid convergence problems and complete the integration successfully. For all these runs the dependent variables were printed whenever $(t_{n+1} - t_n) \geq t_p$, where t_p was a preselected constant.

Each problem was also solved with no printed output but with the purpose of determining the computational time for an error allowance of 1×10^{-5}; to accomplish this we used the subroutine FSTIME [12] which has a negligible overhead of eight microseconds. The time comparison was carried out considering only the double precision arithmetic algorithms. This work was done in the Engineering Systems Simulation Laboratory using an IBM/360-44 computing system.

The results of the different methods for each problem are presented in a table which includes for each error allowance, the amount of steps taken to perform the integration, the number of times that the system of derivatives were evaluated, and, for the error allowance of 1×10^{-5}, the integration time for the IBM/360-44 system.

If the amount of steps and the range of the integration are known, the average stepsize can be calculated. The number of derivative evaluations is important since the computational time is strongly dependent upon this, particularly in the cases where the expressions representing the differential equations involve many arithmetic operations.

The amount of steps was not determined for DESUB because this would have required a modification of the structure of this system. In addition, since DESUB is a modified rational extrapolation algorithm, the amount of steps taken by DESUB will closely parallel the steps taken by the other rational extrapolation algorithm, DIFSUB.

Failure of solution for certain cases was due to either of the following reasons

a) Premature stoppage of the run, due to program "blow up" (numerical instability).

b) The requested error allowance was smaller than that which could be handled by the method.

In the tables presenting the results, "a" is indicated as "unstable", while "b" in indicated as "no convergence".

For a non-linear system the eigenvalues are a function of the state variables, whenever reported these were calculated at certain values of the independent variable using the Jacobian at the same point.

3.1 A System of Reaction Rate Equations

The concentrations of the reactants in a complex chemical reaction system are given by a set of first order differential equations defining the rates of change of concentration with time as functions of the concentration only. Thus a typical set may be written:

$$\frac{d\bar{y}}{dt} = \bar{f}(\bar{y}) \ . \qquad (57)$$

Seinfeld, Lapidus and Hwang [3] solved the following system:

$$\frac{dy_1}{dt} = -0.04y_1 + 10^4 y_2 y_3 \qquad (58)$$

$$\frac{dy_2}{dt} = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2 \qquad (59)$$

$$\frac{dy_3}{dt} = 3 \times 10^7 y_2^2 \qquad (60)$$

with initial conditions

$$y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0$$

where y_1, y_2 and y_3 represent the mole fractions of the three species and t is the independent variable, time.

This problem is a very stiff set of non-linear ordinary differential equations, the eigenvalues of which are initially $\lambda_1 = 0$, $\lambda_2 = 0$, $\lambda_3 = -.04$, but at $t = 0.02$ $\lambda_3 = -2405$.

Figure 3.1 shows a plot of the solution for this system. The behavior of each algorithm in the solution of this problem in the range from t = 0 to t = 15 is shown in Table 3.1.

In the solution of this problem, GEARSB has shown an absolute superiority over the other subroutines because the stiff algorithms in such a program were stable in all three different error allowances, while the rest of the programs were unstable at an error allowance of 1×10^{-3} or less; the polynomial extrapolation algorithm was also unstable at an error of 1×10^{-5}. In addition, the Adams predictor-corrector algorithm in GEARSB was unstable at an error of 1×10^{-3} but it showed stability at an error of 1×10^{-4}. With respect to the execution time, GEARSB stiff method showed a superiority ranging approximately from 1:300 to 1:1000 of the time taken by the other methods. It is interesting to note that Robertson [19] predicts the difference between the best and the worst computation time in the solution of this problem to be a factor of 1000. The single precision version of DESUB was very inefficient for an error of 1×10^{-7} and it required more function evaluations than the double precision version. The slight difference in computation time between the two algorithms for stiff systems from GEARSB is due to the fact that additional function evaluations had to be made in order to numerically approximate the partial derivatives.

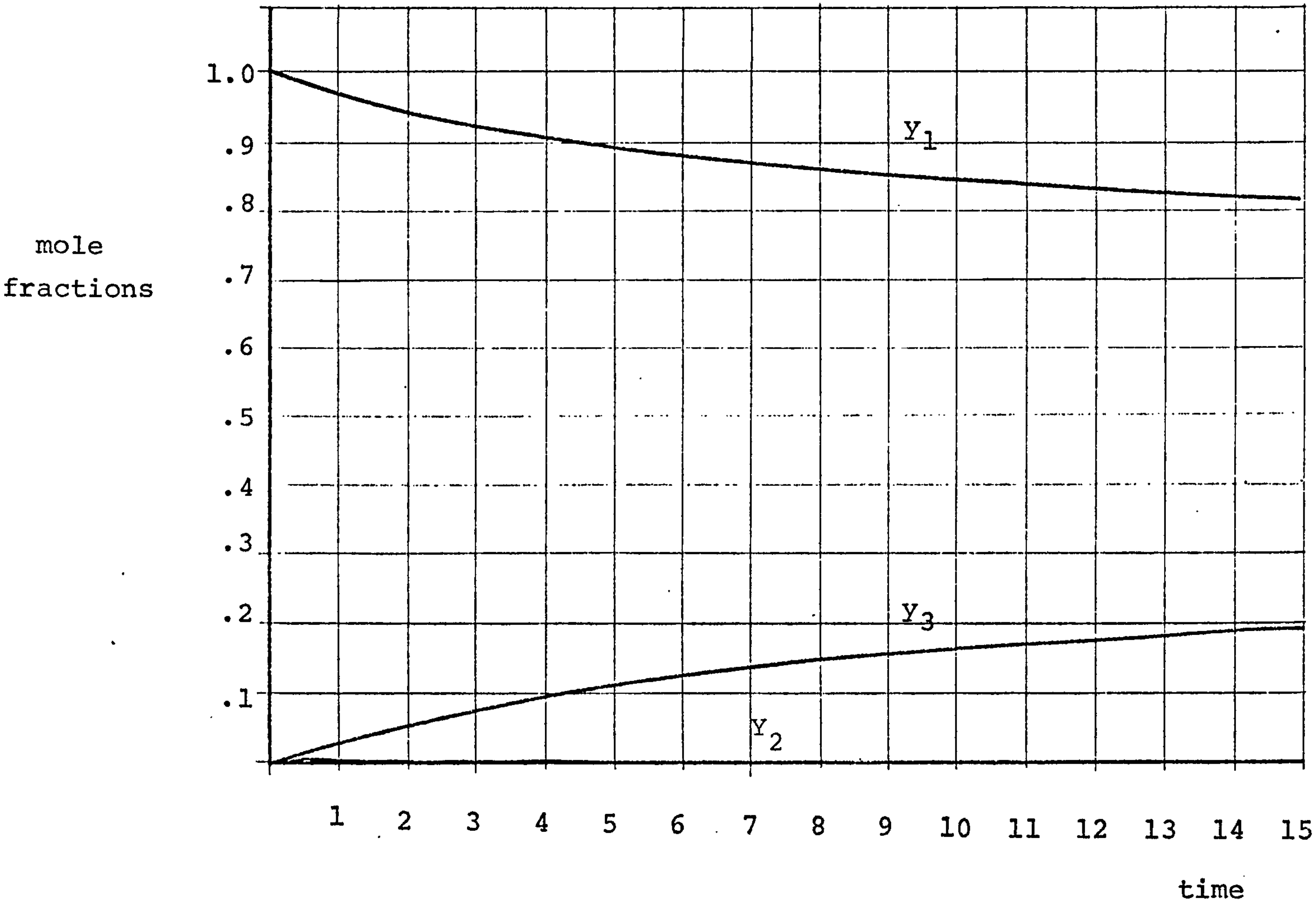


Figure 3.1 A System of Reaction Rate Equations

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	a		6915	211181	498.58	6969	215573
	Polynomial Extrapolation	a		a			6189	190121
DESUB	Double Precision	a			395225	1084.89		361210
	Single Precision	a			523733	1009.50		87191*
GEARSB	Adam's Predictor-Corrector	34217	104926†	32674	99482	533.15	38377	119791
	Gear's Stiff Option 1	31	104	47	126	1.14	92	222
	Gear's Stiff Option 2	27	126	47	162	1.19	92	264
Runge-Kutta	Runge-Kutta-Merson	a		26783	133915	391.27	38347	191735

Table 3.1 A System of Reaction Rate Equations
Integration from t = 0 to t = 15

a unstable
*This integration was performed only from 0 to 1.5
†error = 10^{-4}

3.2 A Stirred Tank Reactor

This development is similar to that of Aris and Amundson [20], which was numerically integrated by Shannon [13] using his changeable independent variable technique. For the case of a non-adiabatic stirred tank reactor with an exothermic first order irreversible reaction occuring with rate

$$r_A = Ac_A \exp\left(-\frac{E}{RT}\right). \tag{61}$$

For a specific numerical example, Shannon [13] shows the following equations:

$$\frac{d\xi}{dt} = 1 - \xi - \xi \exp 50\left(\frac{1}{2} - \frac{1}{\eta}\right) \tag{62}$$

$$\frac{d\eta}{dt} = -2(\eta - 1.75) - 30(\eta - 1.75)(\eta - 2.0) + \xi \exp 50\left(\frac{1}{2} - \frac{1}{\eta}\right) \tag{63}$$

where $\xi = \frac{X_A}{X_{A0}}$

X_A = mole fraction of A in the reactor

X_{A0} = mole fraction of A in the feed stream

$\eta = \frac{C_p T}{X_{A0}(\Delta H)}$

C_p = specific heat of reacting mixture, Btu/mole°R

T = temperature in the reactor, °R

with the initial conditions

$$\xi(0) = 1.0,\ \eta(0) = 1.75.$$

At t = 0 the eigenvalues of the system are $\lambda_1 = 5.9572$, $\lambda_2 = -1.02625$ and at t = 10, $\lambda_1 = -2.625 + 2.4225i$, $\lambda_2 = -2.625 - 2.4225i$.

Figure 3.2 shows the solution of the system while in Table 3.2 the behavior of the different algorithms can be observed.

The behavior of the algorithms in the solution of this system can be summarized as follows:

GEARSB stiff algorithm was again the best, however in this case the computation time ratio among this and the slowest one (DESUB) was of only 1:1.35.

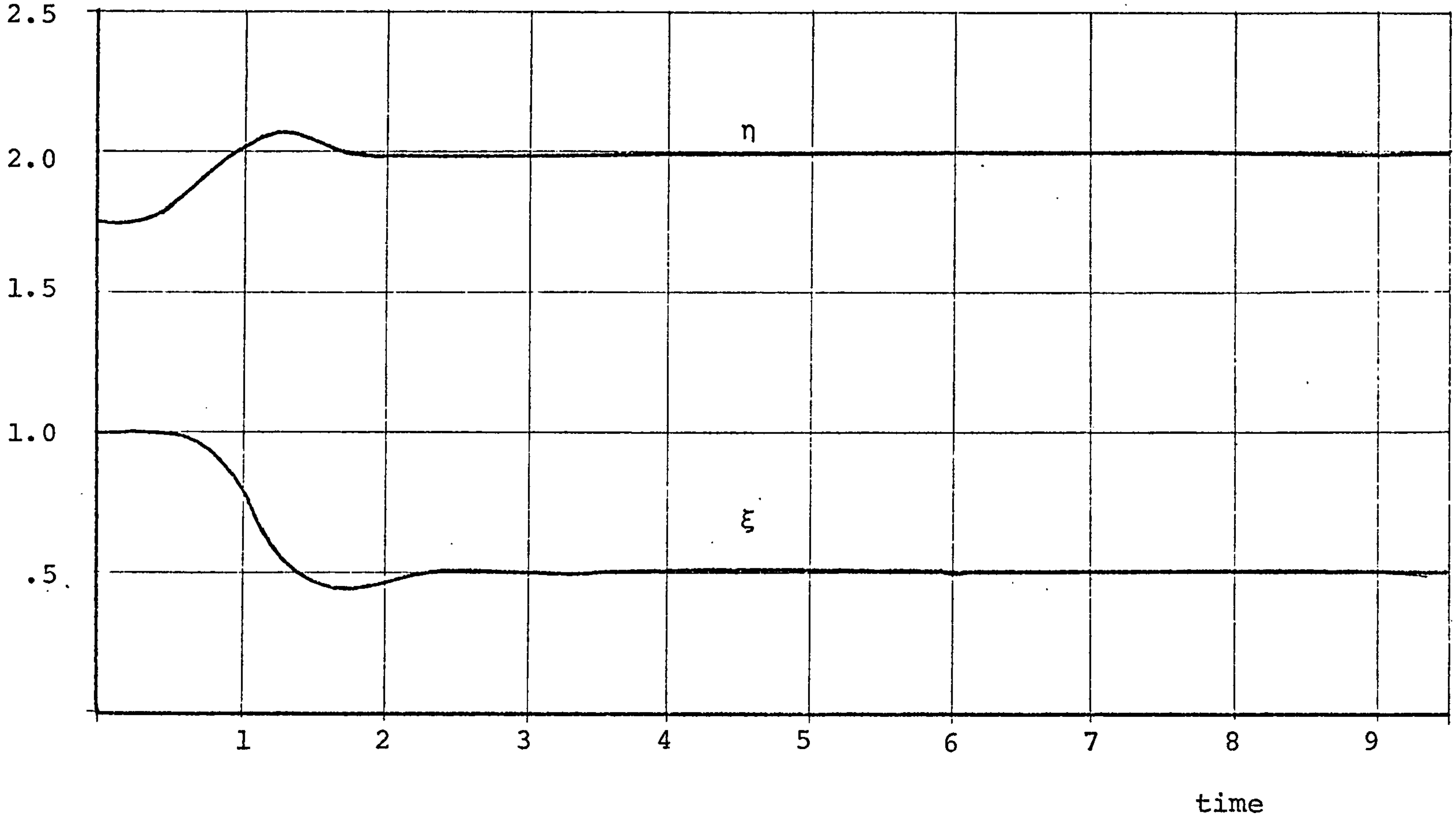


Figure 3.2 A Stirred Tank Reactor

Program	Algorithm	Error = 1×10^{-3} Steps	Func. Eval.	1×10^{-5} Steps	Func. Eval.	Time Secs.	1×10^{-7} Steps	Func. Eval.
DIFSUB	Rational Extrapolation	32	736	40	984	2.55	58	1552
DIFSUB	Polynomial Extrapolation	32	736	38	930	2.245	54	1406
DESUB	Double Precision		783		783	2.86		1071
DESUB	Single Precision		627		851	1.88		2348*
GEARSB	Adam's Predictor-Corrector	98	246	165	382	2.63	241	577
GEARSB	Gear's Stiff Option 1	45	135	105	290	2.133	172	460
GEARSB	Gear's Stiff Option 2	43	148	105	310	2.133	172	484
Runge-Kutta	Runge-Kutta-Merson	66	330	136	810	2.31	367	1835

Table 3.2 The Stirred Tank Reactor Problem
Integration from t = 0 to t = 10

*error = 10^{-6}

3.3 A Turbulent Boundary Layer Problem

Bullin [14] shows the momentum equation for a two-dimensional boundary layer problem on a flat plate for an incompressible fluid to be:

For the inner region

$$f''' = \frac{-ff'' + \lambda[1-(f')^2] - \beta\eta(f'')^2\left[1+\left(\frac{\phi\eta}{\eta_\delta}-1\right)e^{\frac{-\phi\eta}{\eta_\delta}}\right]\left(1-e^{\frac{-\phi\eta}{\eta_\delta}}\right)}{\left(1+2\frac{E}{\nu}\right)}$$

$$+ 2Re\left[f'\left(\frac{\partial f'}{\partial Re}\right)_\eta - f''\left(\frac{\partial f}{\partial Re}\right)\right] \Big/ \left(1 + 2\frac{E}{\nu}\right) \tag{64}$$

For the outer region

$$f''' = \frac{-ff'' + \lambda[1-(f')^2] + 2Re\left[f'\left(\frac{\partial f'}{\partial Re}\right) - f''\left(\frac{f}{\partial Re}\right)_\eta\right]}{\left(1+\frac{E}{\nu}\right)} \tag{65}$$

with boundary conditions

$$\text{at } \eta = 0 \begin{cases} f = 0 \\ f' = 0 \end{cases} \qquad \text{at } \eta \to \infty \begin{cases} f' \to 1 \\ f'' \to 0 \end{cases}$$

where f = dimensionless similarity variable $\left(f' = \frac{u}{u_o}\right)$

f' = derivative of f with respect to η at constant x and λ

η = dimensionless similarity variable $= \left[\frac{1}{2\nu u_a \xi}\right]^{\frac{1}{2}} u_o y$

$\lambda = -\frac{2\xi}{u_o}\frac{du_o}{d\xi}$

ξ = independent similarity variable in x-direction, ft.

$= \int_o^x \frac{u_o}{u_a}\,dx$

u_a = streamwise approach velocity, ft/sec

u_o = streamwise velocity at the outer edge of boundary layer

ν = kinematic viscosity, lbs/ft^2

β = constant = $2k^2(2Re)^{1/2}$

k = constant in Eddy viscosity equation, dimensionless

Re = Reynold's number = $\frac{u_a \xi}{\nu}$

η_δ = value of η at outer edge of boundary layer, dimensionless

ϕ = $\frac{y_m^+ - a}{b}$

y_m^+ = maximum value of y^+

a = turbulent damping factor

b = constant

y^+ = distance in y-direction in terms of wall law, dimensionless

E = Eddy viscosity, lbs/ft^2.

As can be seen, the equation is a non-linear, third order ordinary differential equation. In order to solve such a problem, we transformed it into an equivalent first order system of three equations.

$$f_1 = f' \tag{66}$$

$$f_2 = f'' \tag{67}$$

$f_3 = f'''$ [given in equations (64) and (65)]

Because we were only interested in the numerical integration of such a system, and not in the solution of the two point boundary value problem, we used the initial value of f'' as calculated by Shannon [13], as well as the coefficients calculated by the linear regression analysis subroutine written by Shannon.

The numerical values of the different parameters we used were:

$a = 1.0$

$b = 25.0$

$\eta_\delta = 34.42$

$k = 0.4$

$\lambda = 0.$

$Re = 1 \times 10^7$

and the initial conditions:

$f(0) = 0,$

$f_1(0) = 0,$

$f_2(0) = 5.4$

at $\eta = 0$ the eigenvalues of the system are $\lambda_1 = -1.7544$, $\lambda_2 = -1.7544$ and $\lambda_3 = -1.7544$.

The shape of the solution is presented in Figures 3.3.1 and 3.3.2 and the summary of the performance of the algorithm is in Table 3.3.

Runge-Kutta was unstable with an error allowance of only 10^{-3}. The single precision arithmetic version of DESUB presented a limiting error of 5.58×10^{-6} and therefore the problem was not solved for an error allowance of 1×10^{-7} with this method while the GEARSB stiff algorithm could only be solved for an error allowance of 1×10^{-6}, however the Adams predictor-corrector algorithm of GEARSB program had the best computation time showing a superiority ranging from 1:1.06 to 1:1.79.

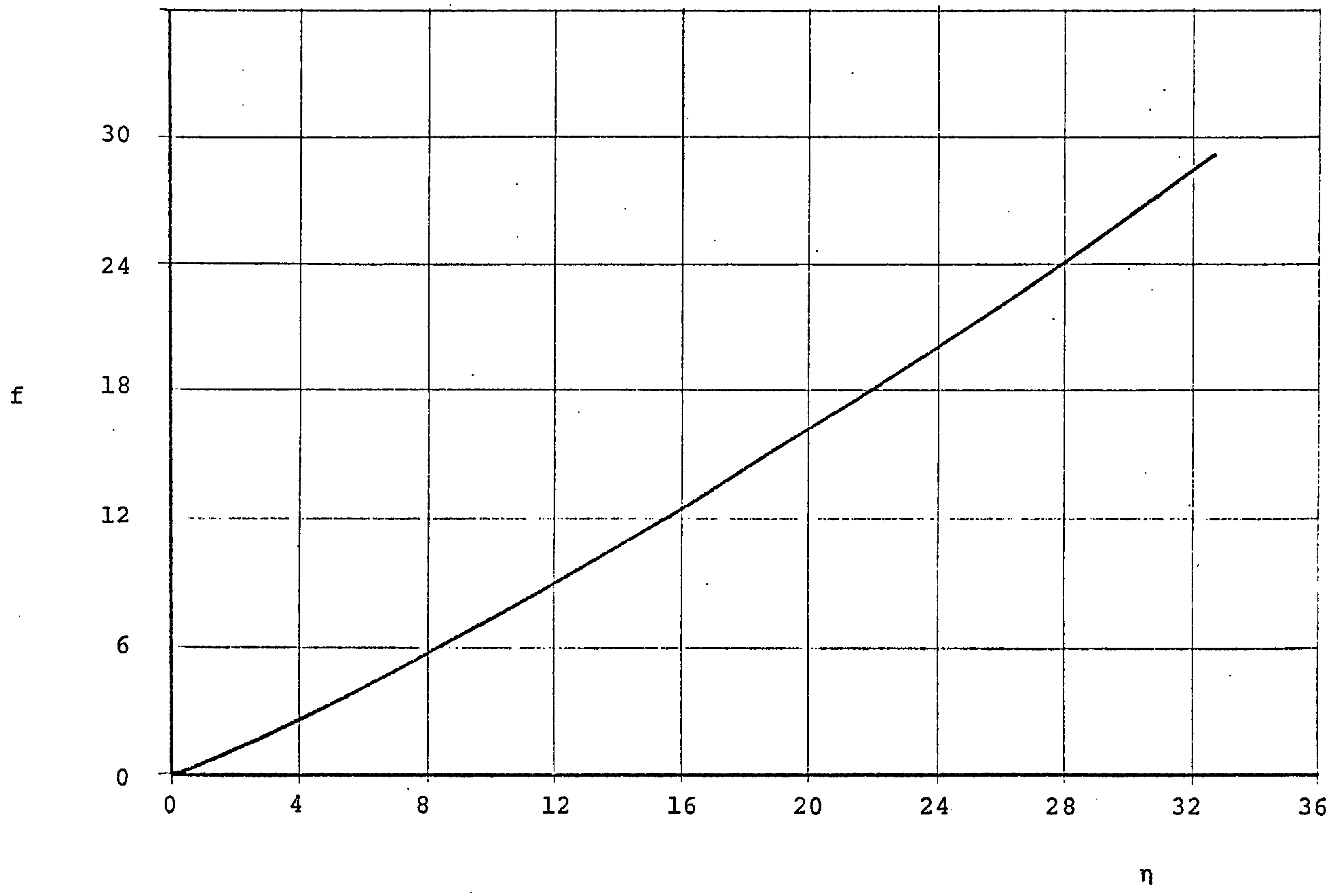


Figure 3.3.1 Turbulent Boundary Layer

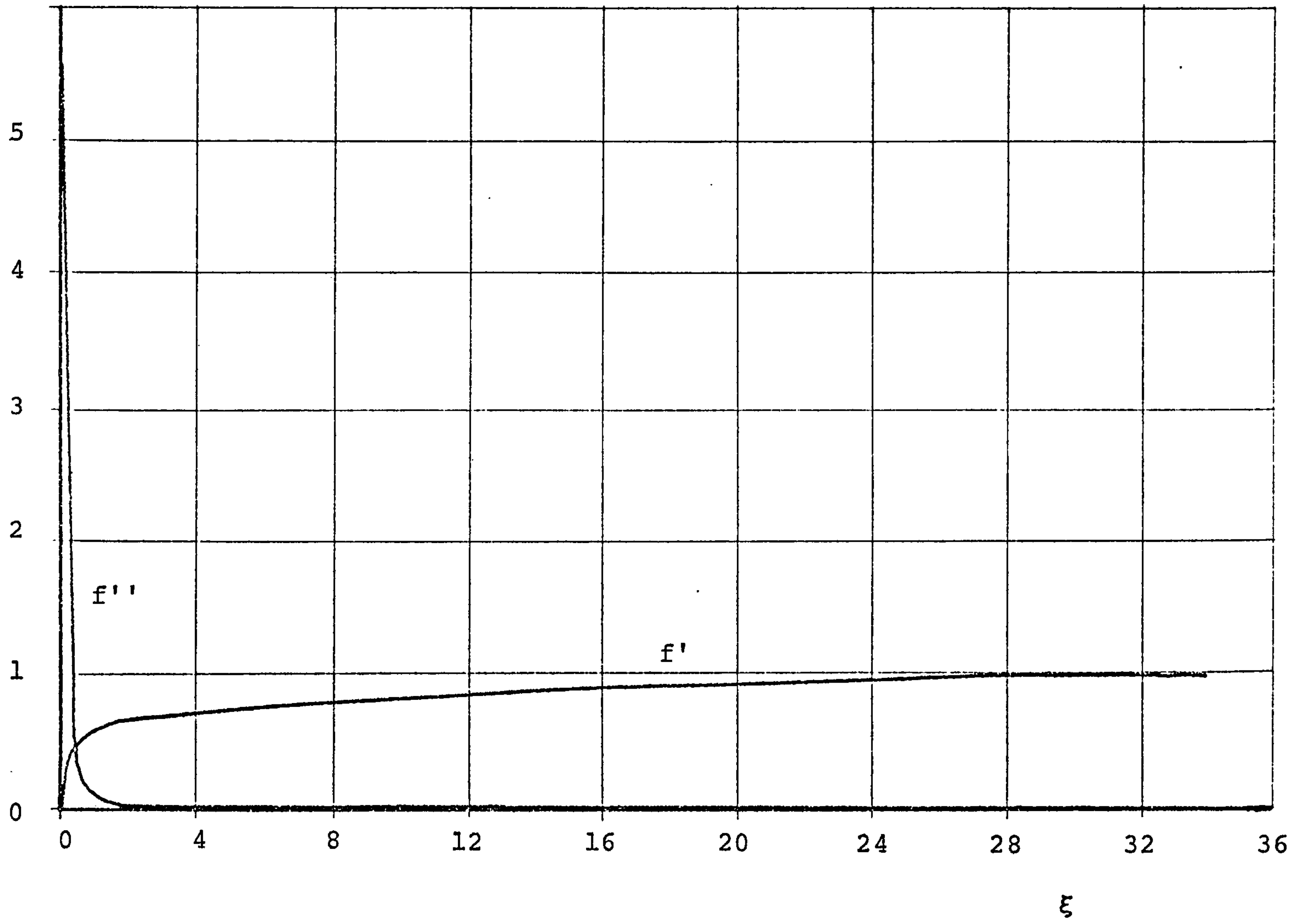


Figure 3.3.2 Turbulent Boundary Layer

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	30	630	40	944	10.12	57	1469
	Polynomial Extrapolation	33	725	38	886	9.197	54	1370
DESUB	Double Precision		463		1005	11.83		2193
	Single Precision		475		561	5.02	b	
GEARSB	Adam's Predictor-Corrector	99	450	164	786	6.63	224	1252
	Gear's Stiff Option 1	70	410	127	794	7.05	205	1240*
	Gear's Stiff Option 2	67	458	125	910	7.69	205	1340*
Runge-Kutta	Runge-Kutta-Merson	a		158	790	8.29	453	2265

Table 3.3 The Turbulent Boundary Layer

Integration from $\eta = 0$ to $\eta = 34.42$

a unstable
b no convergence
*error = 10^{-6}

3.4 Periodic Chemical Reaction Problem

Sincic [18] has investigated forced oscillation in a chemical reaction using the inlet temperature of the feed stream as a forcing function. Two forms of forcing functions were considered: sinusoidal and "bang-bang" (step functions); in both cases the middle value of the function was the same.

The following system with specified numerical values was selected:

$$\frac{dC}{dt} = 6.5 - C - 1.75754943 \times 10^{14} \exp(-14088/T)C \quad (68)$$

$$\frac{dT}{dt} = 2.162(T_0 - T) + 4.745383461 \times 10^{15} C \exp(-14088/T) \quad (69)$$

where C = concentration, moles/liter

T = temperature, °K

t = time, dimensionless.

For the sinusoidal forcing function:

$$T_0 = 390 + 10\cos(2\pi t/p)$$

where p = period = 6.

For the bang-bang forcing function:

$$T_0 = \begin{cases} 400 \text{ for } 6n + \frac{6}{4} \leq t \leq 6n + \frac{18}{4} \\ 380 \text{ for } 6n \leq t \leq 6n + \frac{6}{4} \\ \qquad \text{and for } 6n + \frac{18}{4} \leq (n+1)6 \end{cases}$$

where n = 0, 1, 2, ...

It is interesting to notice that while the sinusoidal forcing function may vary the value of the parameter "T_0" smoothly, the bang-bang makes it vary sharply and allows only two values; this factor obviously is cause of additional stiffness in the numerical integration.

At t = 0, the eigenvalues of the system are: $\lambda_1 = -.3804 + .7320i$, $\lambda_2 = -.3804 - .7320i$; while at $t = 1.57$, $\lambda_1 = -.1926645$, $\lambda_2 = -.5944355$ and at $t = 18.282$, $\lambda_1 = .4097$, $\lambda_2 = .6798$.

Table 3.4.A shows that for the case of the sinusoidal forcing function, the three algorithms of GEARSB had very similar behavior and all three were superior to the rest. The slowest one was the Runge-Kutta, which took 3.83 times the time required by option one of GEARSB stiff algorithm. The rational extrapolation and polynomial extrapolation algorithms form DIFSUB took respectively only 1.97 and 1.53 times more time than GEARSB stiff algorithm.

Figure 3.4.A shows the plot of the solution of this problem.

For the case of the bang-bang forcing function, it is interesting to notice that stiff algorithms from GEARSB took less time in performing the integration than the others, but they had convergence problems at small error allowances of 1×10^{-5}; however, the system could not be solved by this algorithm for an error allowance of 1×10^{-7}. Although Runge-Kutta and DIFSUB algorithms lacked convergence problems for small error allowances, Runge-Kutta required more time by a factor of eight than the stiff algorithm from GEARSB, and rational extrapolation and polynomial extrapolation from DIFSUB required the same by a factor of 4.77 and 4.25 respectively.

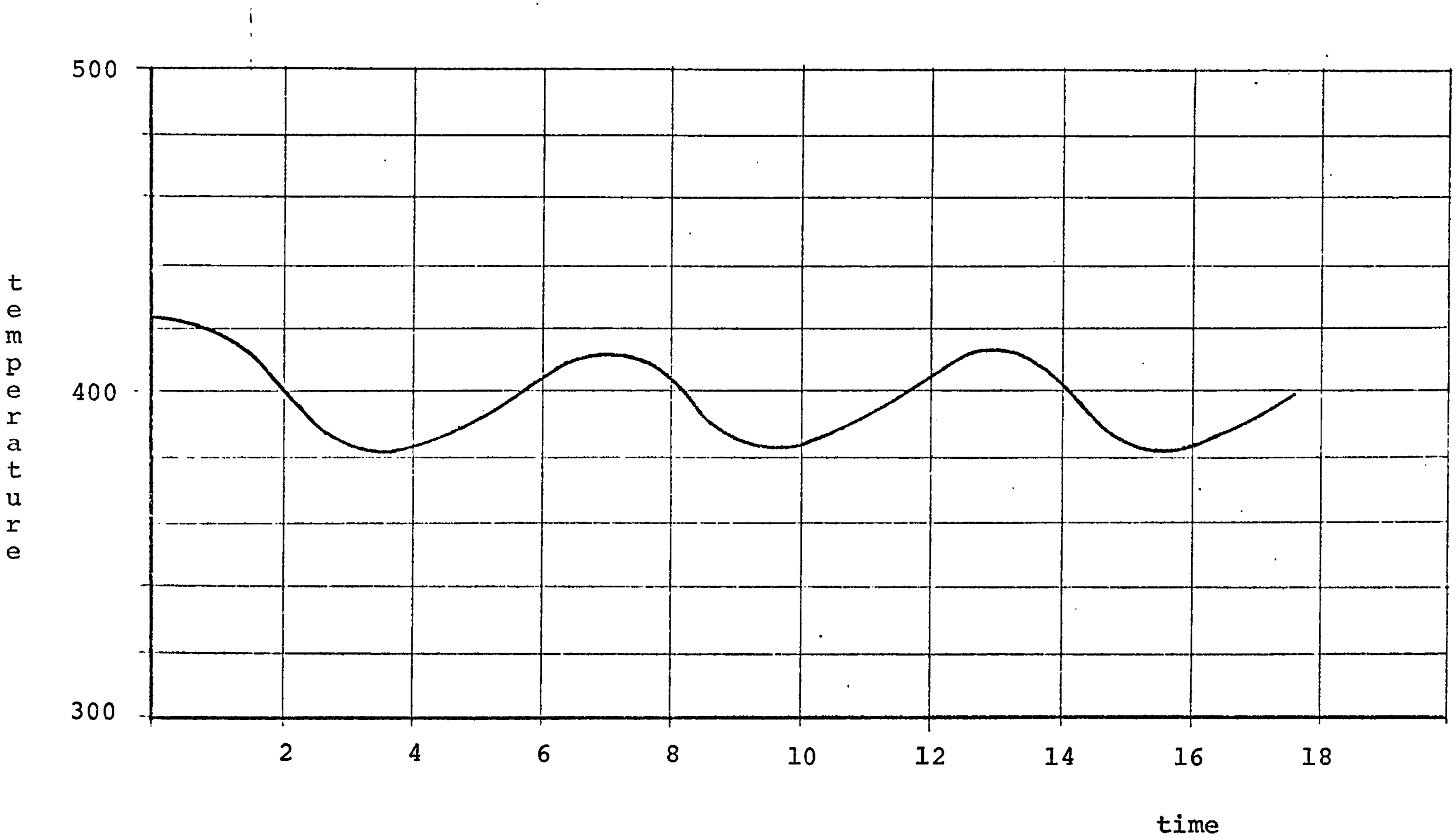


Figure 3.4.1.A Periodic Chemical Reaction (Sinusoidal Forcing Function)

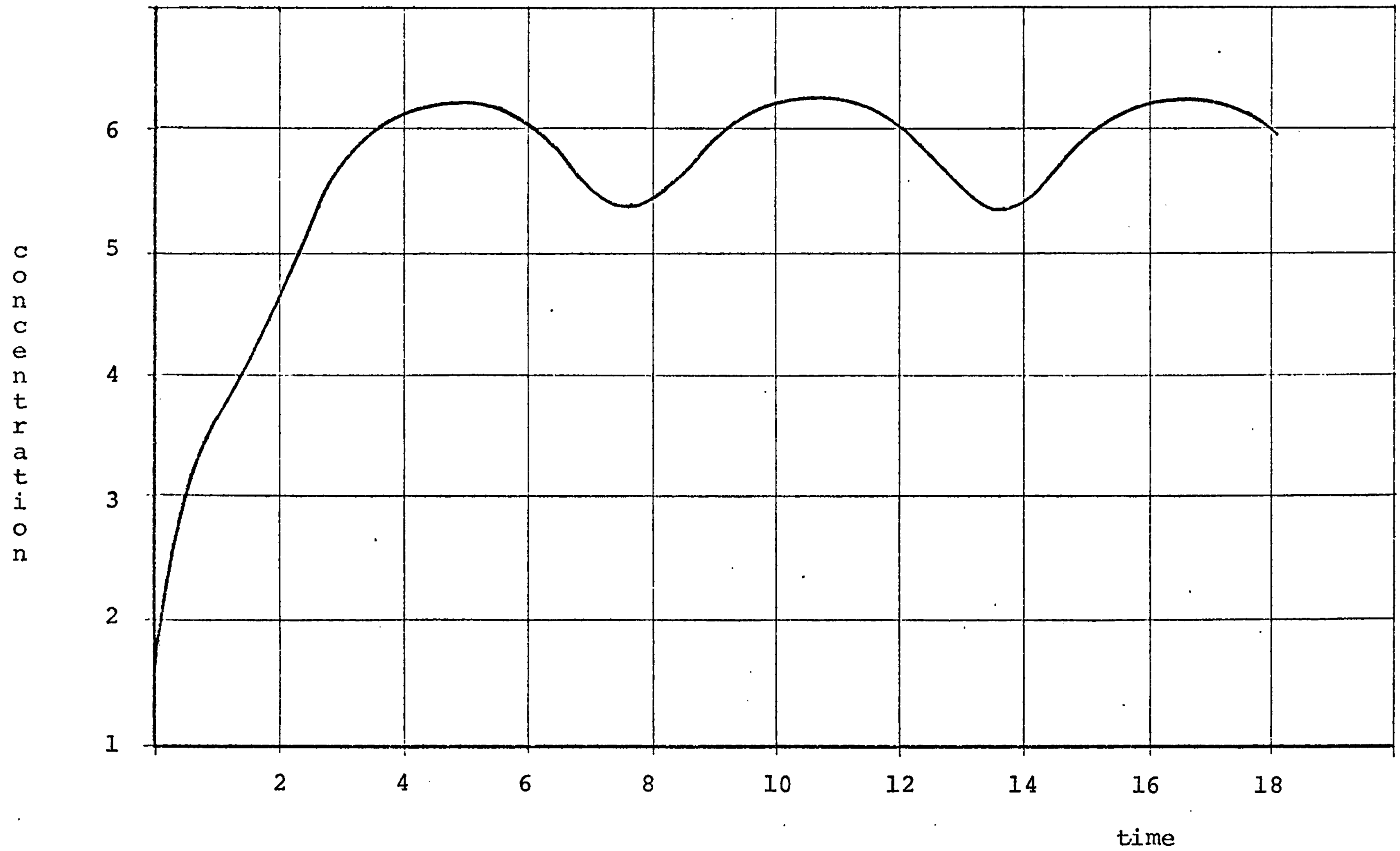


Figure 3.4.1.B Periodic Chemical Reaction (Sinusoidal Forcing Function)

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	37	889	48	1232	7.855	75	2071
DIFSUB	Polynomial Extrapolation	36	868	46	1174	6.105	70	1918
DESUB	Double Precision	a			963	4.948	a	
DESUB	Single Precision	a			a		b	
GEARSB	Adam's Predictor-Corrector	115	281	201	493	4.195	293	738
GEARSB	Gear's Stiff Option 1	65	235	141	442	3.986	217	669
GEARSB	Gear's Stiff Option 2	69	318	141	476	4.043	217	745
Runge-Kutta	Runge-Kutta-Merson	790	3958	728	3640	15.283	2246	11230

Table 3.4.A Periodic Chemical Reaction (Sinusoidal Forcing Function)
Integration from $t = 0$ to $t = 18$

a unstable
b no convergence

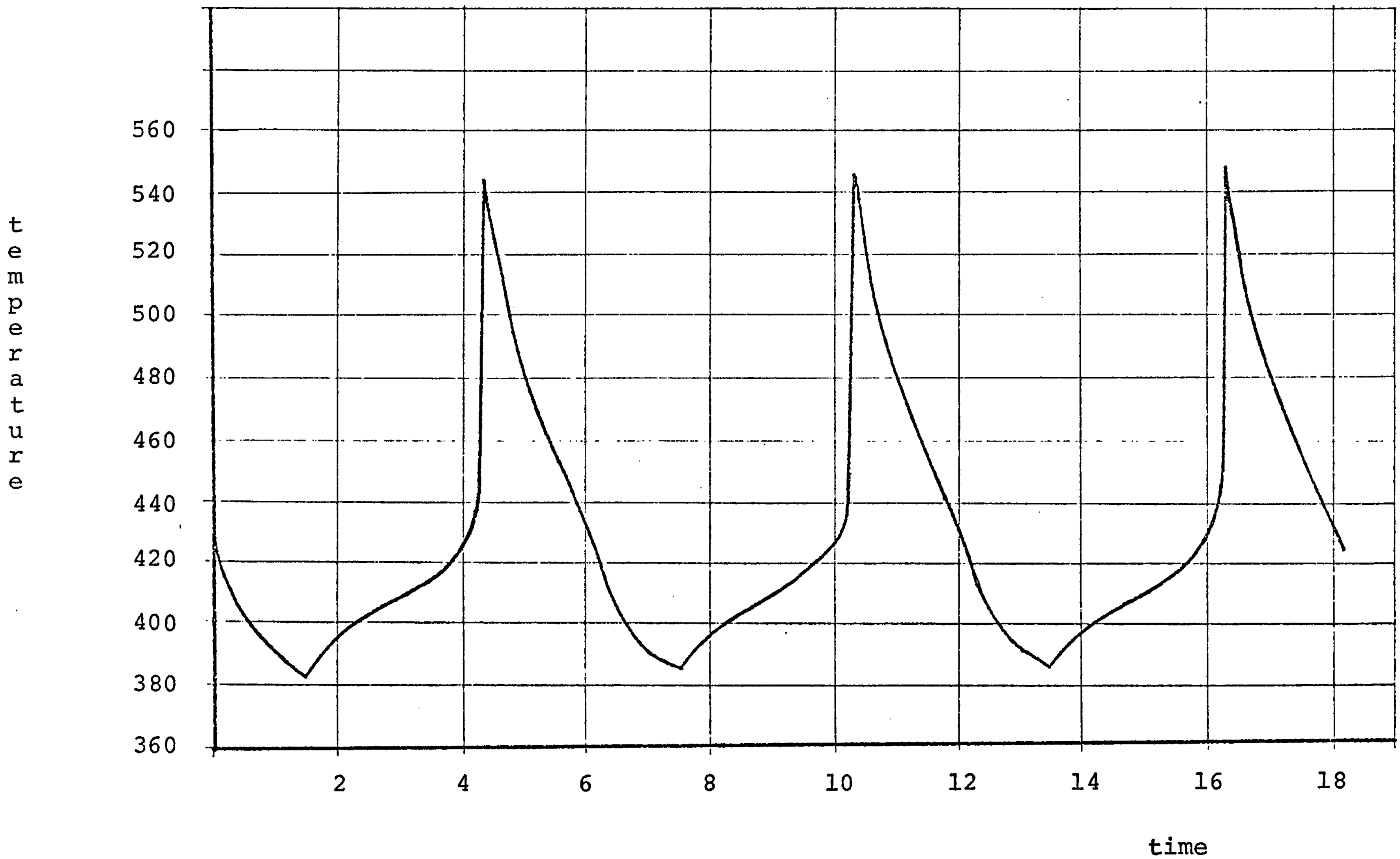


Figure 3.4.2.A Periodic Chemical Reaction (Bang-bang Forcing Function)

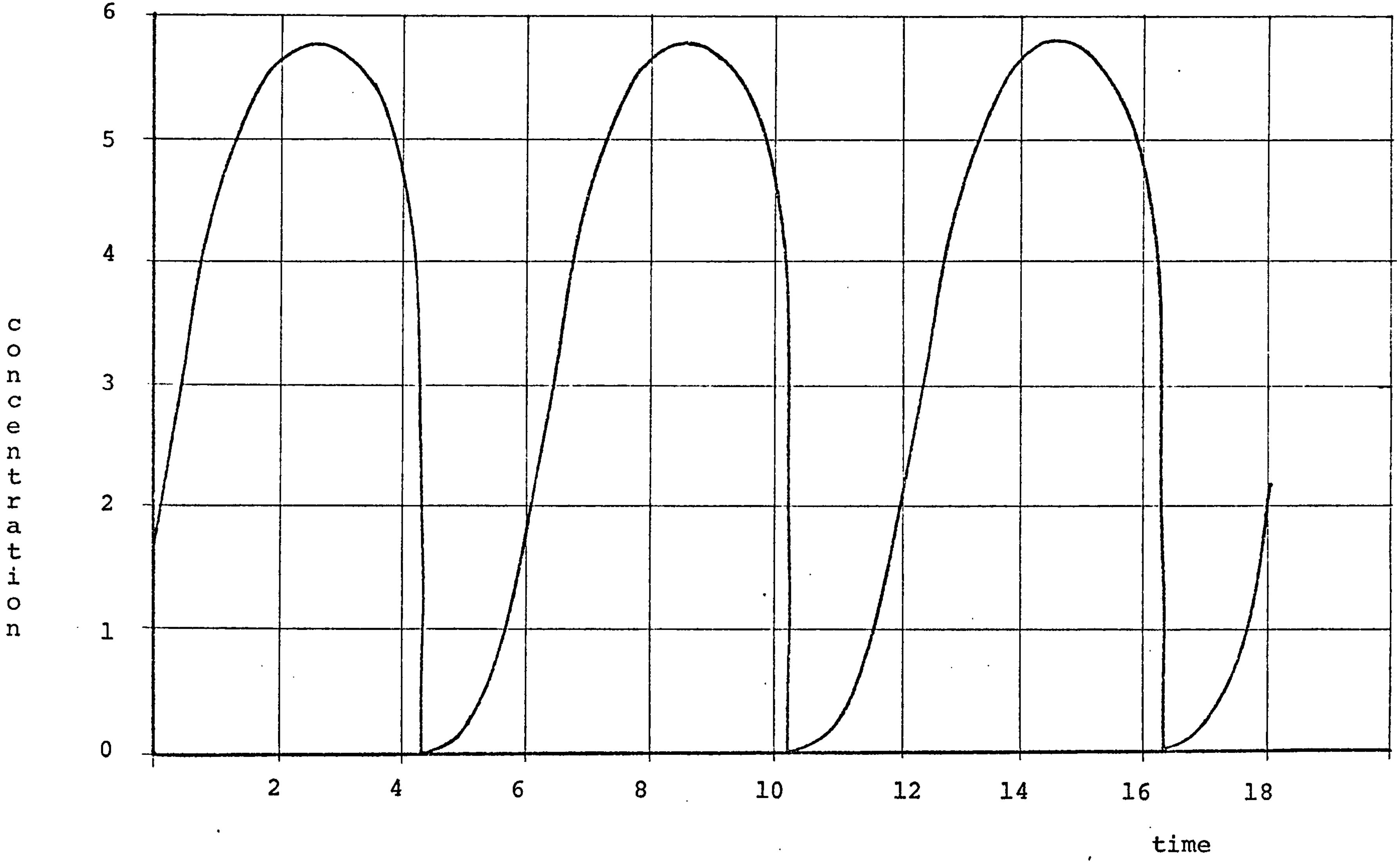


Figure 3.4.2.B Periodic Chemical Reaction (Bang-bang Forcing Function)

Program	Algorithm	Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
		Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	197	5639	306	9128	38.5236	545	16537
	Polynomial Extrapolation	178	5064	285	8509	34.3618	517	15709
DESUB	Double Precision	a		a			a	
	Single Precision	a		a			b	
GEARSB	Adam's Predictor-Corrector	1007	2800	1374	3662	29.02	b	
	Gear's Stiff Option 1	210	802	234	914	8.07*	b	
	Gear's Stiff Option 2	220	1094	232	1157	8.3573*	b	
Runge-Kutta	Runge-Kutta-Merson	1197	55985	3246	16230	68.7887	9678	48390

Table 3.4.B Periodic Chemical Reaction (Bang-bang Forcing Function)
Integration from t = 0 to t = 18

a unstable
b no convergence
*The error allowance had to be automatically increased to an average of 5×10^{-4}

3.5 The Thermal Decomposition of Ozone

This problem was used by Bowen et al. [16] as an important system in singularily perturbed form with which to derive an analytical solution approximation via matched asymptotic expansions. The accepted kinetic steps involved for a dilute ozone oxygen mixture are

$$O_3 + O_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} O + 2O_2$$

$$O_3 + O \xrightarrow{k_3} 2O_2 .$$

Aiken and Lapidus [15] defined the following dimensionless variables

$$y = [O]/E[O_3]_0 \tag{70}$$

$$x = [O_3]/[O_3]_0 \tag{71}$$

$$k = 2k_2[O_2]_0/k_1 \tag{72}$$

$$E = k_1[O_2]_0/2k_3[O_3]_0 \tag{73}$$

where $[O]_0$ = initial concentration of oxygen atom

$[O_2]_0$ = initial concentration of oxygen molecule

$[O_3]_0$ = initial concentration of ozone

$[O]$ = concentration of oxygen atom

$[O_2]$ = concentration of oxygen molecule

$[O_3]$ = concentration of ozone

and the time scale divided by $2/k_1[O_2]_0$. They described the transient behavior by

$$\frac{dx}{dt} = -x - xy + Exy \qquad (74)$$

$$E\frac{dy}{dt} = x - xy - Exy \qquad (75)$$

with initial conditions

$$x(0) = 1, \qquad y(0) = 0.$$

This problem was solved utilizing the following numeric values:

$E = 1.0/98$

$k = 3.0.$

The eigenvalues of this problem at $t = 0$ were $\lambda_1 = 0$, $\lambda_2 = -102$ and at $t = 100$, $\lambda_1 = -2.0137 + 1.4092i$, $\lambda_2 = -2.0137 - 1.4092i$.

In Figure 3.5 we can observe the shape of the solution while in Table 3.5 we can see the behavior of the algorithm in its solution.

For high accuracy results (error allowance of 10^{-7}) the single precision version of DESUB was very inefficient, while for an error allowance of 10^{-3} the double precision version showed instability; however, the problem can be solved with an error allowance of 10^{-4}.

The most efficient method for this problem was the stiff algorithm of GEARSB. Its superiority in time computation ranged from approximately 1:2 if compared with DIFSUB to 1:3.5 if compared with DESUB.

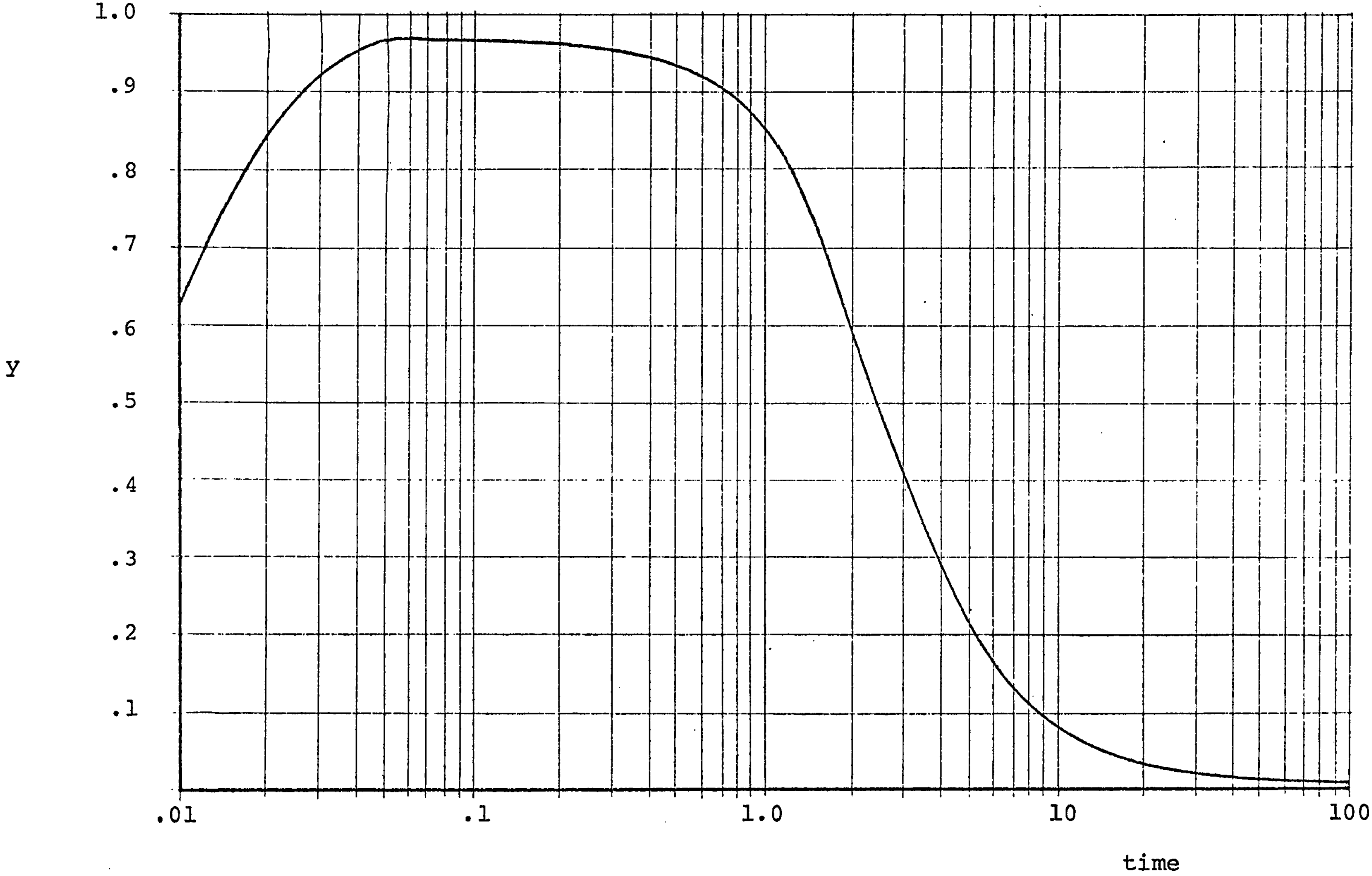


Figure 3.5.1 Thermal Decomposition of Ozone

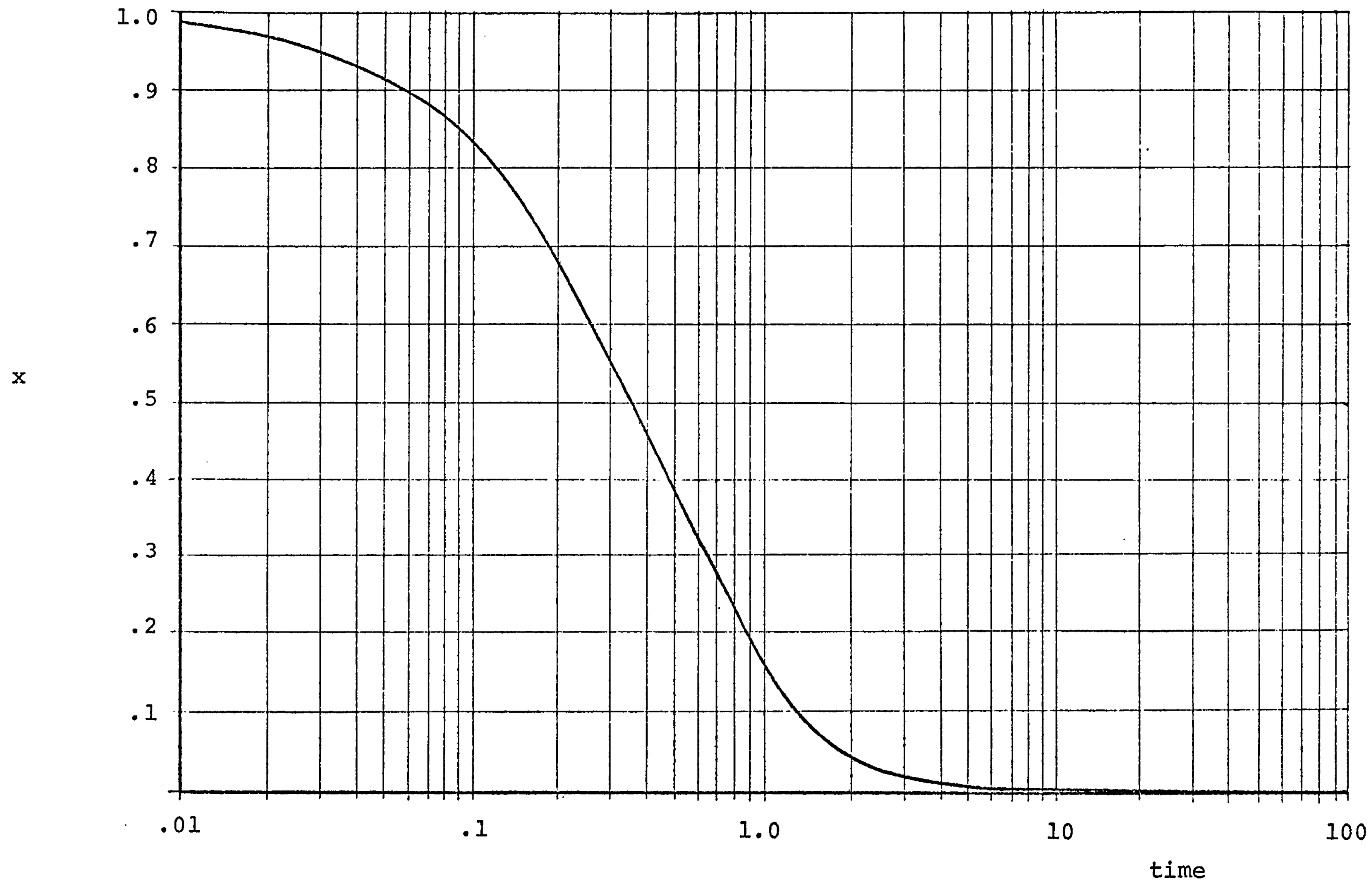


Figure 3.5.2 Thermal Decomposition of Ozone

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	105	3013	163	4167	7.855	206	6038
DIFSUB	Polynomial Extrapolation	98	2758	125	3589	6.105	125	5699
DESUB	Double Precision		5104*		4925	11.828		5467
DESUB	Single Precision		4075		4920	6.93		19504
GEARSB	Adam's Predictor-Corrector	510	1457	746	1891	10.28	1045	2227
GEARSB	Gear's Stiff Option 1	86	232	186	495	3.565	274	698
GEARSB	Gear's Stiff Option 2	86	268	186	539	3.614	274	764
Runge-Kutta	Runge-Kutta-Merson	401	2005	578	2890	8.273	1272	6360

Table 3.5 Thermal Decomposition of Ozone
Integration from t = 0 to t = 100

*error = 10^{-4}

3.6 Transient Behavior of a Catalytic Fluidized Bed

Luss and Amundson [17] have examined a specific model for the dynamics of a catalytic fluidized bed in which mixing is complete and heat and mass transfer resistances are lumped at the particle surfaces. An irreversible gas phase first order reaction (A → B) is assumed to occur within the uniform porous catalyst pellet, each of which is at the same partial pressure and temperature. Aiken and Lapidus [15] solved this problem with the equations and parameters numerically specified as follows

$$\frac{dx}{dt} = 1.30(y_2 - x) + 1.04 \cdot 10^4 ky \qquad (76)$$

$$\frac{dy_1}{dt} = 1.88 \cdot 10^3 [y_3 - y_1(1 + k)] \qquad (77)$$

$$\frac{dy_2}{dt} = 1752 - 269y_2 + 267x \qquad (78)$$

$$\frac{dy_3}{dt} = 0.1 + 320y_1 - 321y_3 \qquad (79)$$

and initial conditions

$$x(0) = 759.167,$$

$$y_1(0) = 0.0,$$

$$y_2(0) = 600,$$

$$y_3(0) = 0.1$$

where $k = 0.0006 \exp(20.7 - 15000/y_2)$

x = temperature of the particles, °R

y_1 = partial pressure of the particles, atm

y_2 = temperature of interstitial fluid, °R

y_3 = partial pressure of the interstitial fluid, atm.

Luss and Amundson [17] report the existence of three possible steady states, and they determined the eigenvalues by solution of the fourth order polynomial by use of the Newton-Raphson method. The values of the eigenvalues for the three steady states are:

	First Steady State	Second Steady State	Third Steady State
λ_1	-0.00632	+0.00610	-0.00898
λ_2	-0.91177	-1.26387	-12.6008
λ_3	-270.3147	-270.731	-270.7236
λ_4	-2186.977	-2183.084	-2258.0596

Since the eigenvalues of these states differ by six orders or magnitude, the system is very stiff.

Figure 3.6 shows the plot of the solution of this problem while in Table 3.6 it can be seen that the algorithms for stiff systems in GEARSB are very superior to the others. Its superiority in computational time for an error allowance of 1×10^{-5} was approximately 1 to 100 if compared with polynomial extrapolation of DIFSUB and 1 to 125 if compared with rational extrapolation of DIFSUB and Runge-Kutta.

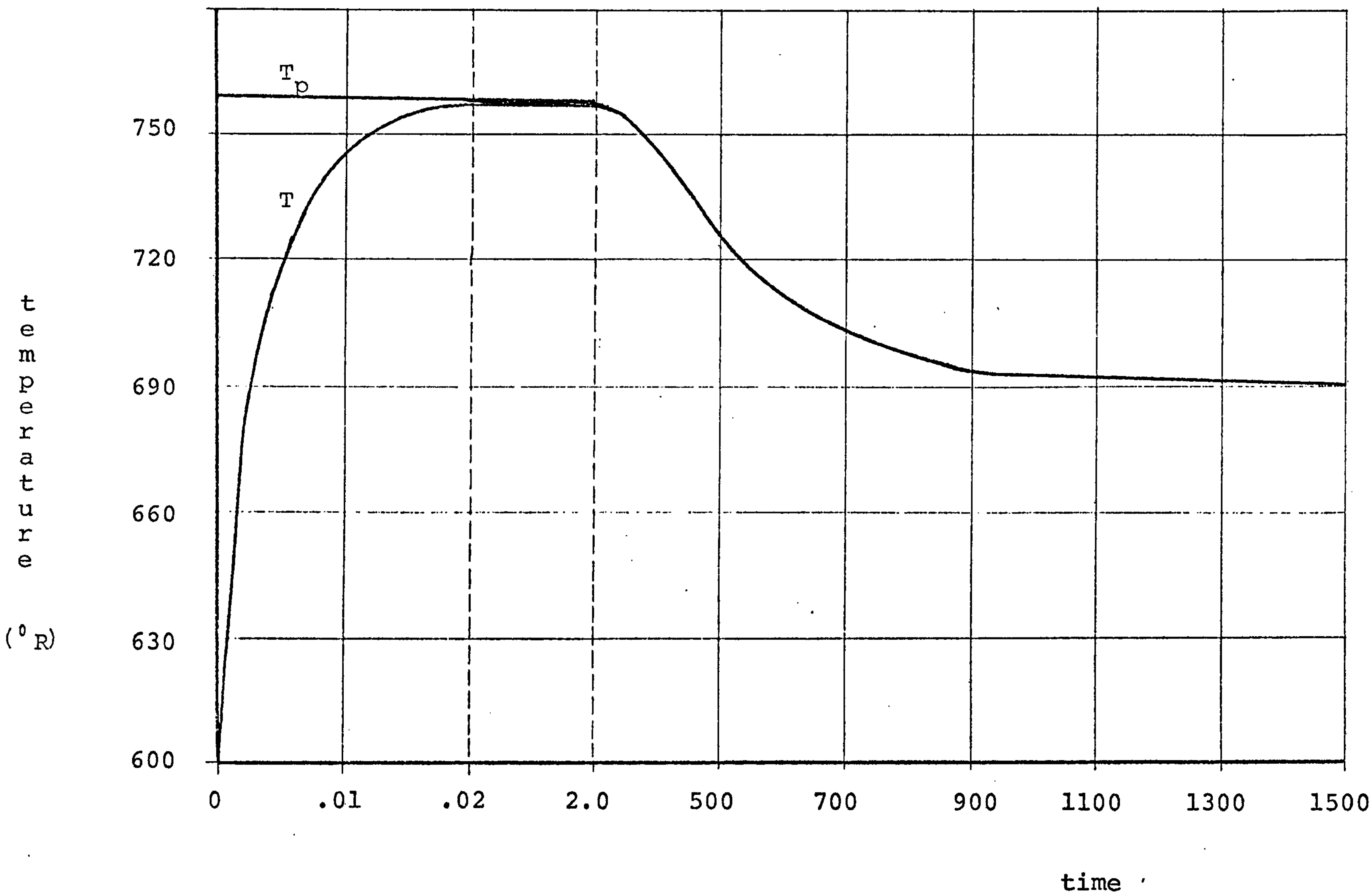


Figure 3.6.1 The Transient Behavior of a Catalytic Fluidized Bed

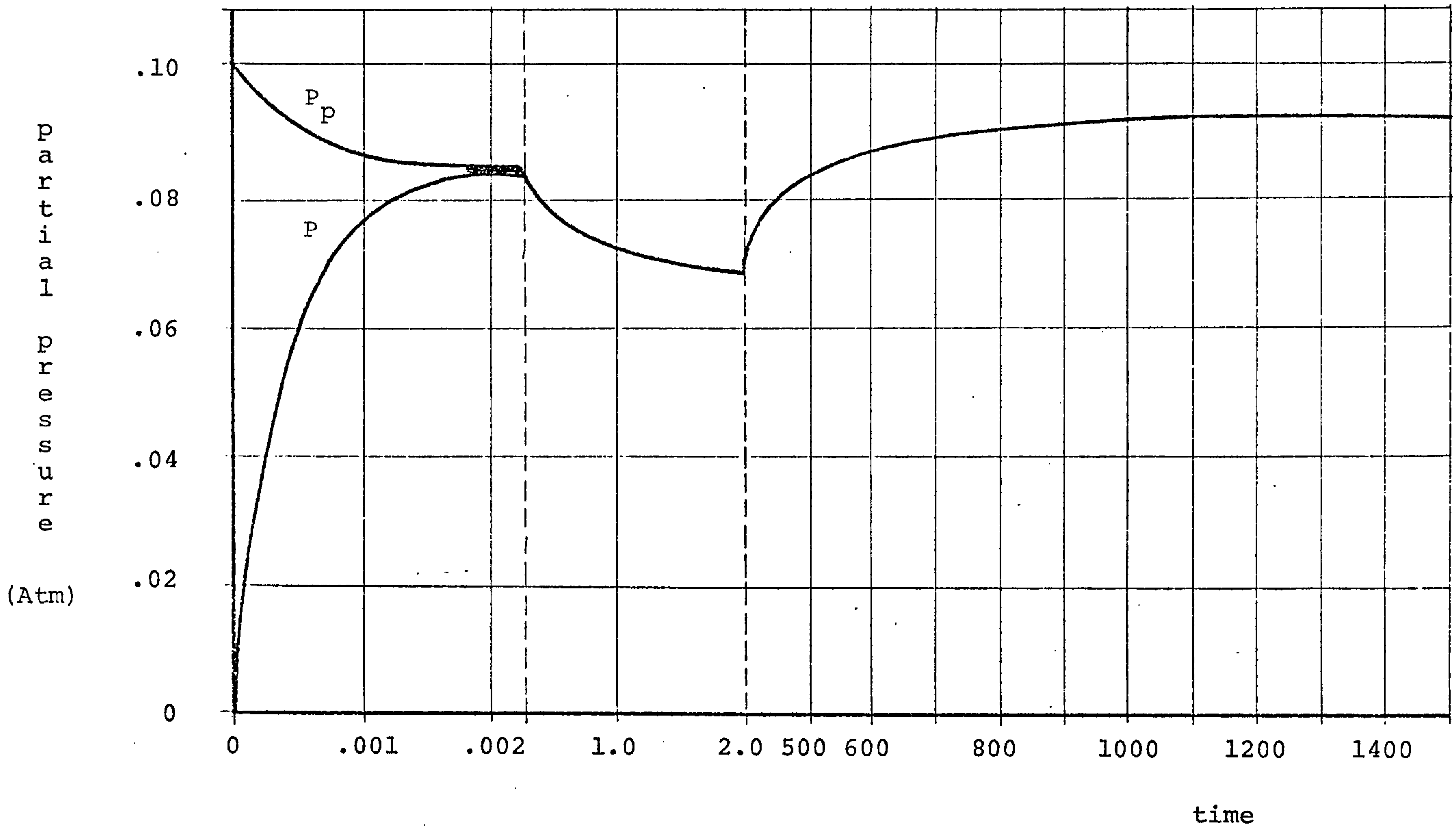


Figure 3.6.2 The Transient Behavior of a Catalytic Fluidized Bed

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	4152	128368	4160	128600	462.55	4168	128780
	Polynomial Extrapolation	3698	113566	3701	113677	366.8	3711	113991
DESUB	Double Precision		a		a			a
	Single Precision		a		a			a
GEARSB	Adam's Predictor-Corrector	18988	61068	19192	61733	439.43	21574	71729
	Gear's Stiff Option 1	45	105	108	268	3.69	205	522
	Gear's Stiff Option 2	45	161	108	336	3.83	205	622
Runge-Kutta	Runge-Kutta-Merson	21332	106660	21475	107375	454.058	21992	109960

Table 3.6 Transient Behavior of a Catalytic Fluidized Bed Integration from t = 0 to t = 10

a unstable
b no convergence

3.7 Mathematical Model of Human Muscle-Chemical Aspects

Bidani and Flumerfelt [21] have developed a mathematical model for the closed loop human respiratory system. The muscle model presented here is one of its subsystems. This lumped model accounts for transport (convective, diffusive and active), storage, depletion and interconversion of the respiratory species (physically dissolved oxygen, oxyhemoglobin, physically dissolved carbon dioxide, bicarbonate ion, hydrogen ion and carbamino).

For details on this development refer to Bidani [21].

Because of blood flow transit delays inherent in the respiratory system, the equations describing it take the form of non-linear differential-difference equations. Solution of such a system proceeds sequentially, each subsystem being integrated in turn.

The limitation imposed on such a sequential integration of the system is that the maximum time step taken for numerical integration of any of the subsystems must be less than the smallest value of the delay time in the model. It is for this reason that the Muscle equations are solved here with time step equal to 0.05 minutes, and for the same reason we decided to use the fixed step Runge-Kutta-Gill algorithm, rather than the variable step Runge-Kutta-Merson used in all the other problems.

This model was solved using only the double precision algorithms.

Equations for Muscle System

Intracellular Fluid

Dissolved carbon dioxide:

$$\left(\frac{dP_{c_{CO_2}}}{dt}\right) = \left(\frac{1}{\alpha_{CO_2} V_c}\right)\left[- D_{csc}\left(P_{c_{CO_2}} - P_{s_{CO_2}}\right) + M_{T_{CO_2}} F - V_c\left(k_u \alpha_{CO_2} P_{c_{CO_2}} - \frac{k_v}{k}[H]_c [HCO_3]_c\right)\right] \tag{80}$$

Oxygen: Physically dissolved and chemically combined with myoglobin

$$\left(\frac{dP_{c_{O_2}}}{dt}\right) = \left(\frac{1}{\bar{\alpha}_{m_{O_2}} V_c}\right)\left[D_{sco}\left(P_{s_{O_2}} - P_{c_{O_2}}\right) - M_{T_{O_2}}\right] \tag{81}$$

Bicarbonate ion:

$$\left(\frac{d[HCO_3]_c}{dt}\right) = \left(\frac{1}{V_c}\right)\left[(1 - F)M_{T_{CO_2}} - D_{csb}(r_{cs}[HCO_3]_c - [HCO_3]_s) + V_c\left(k_u \alpha_{CO_2} P_{c_{CO_2}} - \frac{k_v}{k}[H]_c [HCO_3]_c\right)\right] \tag{82}$$

Hydrogen ion:

$$\left(\frac{d[H]_c}{dt}\right) = \left(\frac{1}{V_c}\right)\left[(1 - F)M_{T_{CO_2}} + V_c\left(k_u\alpha_{CO_2}P_{c_{CO_2}} - \frac{k_v}{k}[H]_c[HCO_3]_c\right)\right.$$

$$\left. - K_{csh}(t_{cs}[H]_c - [H]_s)\right]\left[- \frac{2.303[H]_c}{\beta_c}\right] \qquad (83)$$

Interstitial Fluid

Dissolved carbon dioxide:

$$\left(\frac{dP_{s_{CO_2}}}{dt}\right) = \left(\frac{1}{\alpha_{CO_2}V_s}\right)\left[- D_{spc}\left(P_{s_{CO_2}} - P_{B_{CO_2}}\right) + D_{csc}\left(P_{c_{CO_2}} - P_{s_{CO_2}}\right)\right.$$

$$\left. - V_s\left(k_u\alpha_{CO_2}P_{s_{CO_2}} - \frac{k_v}{k}[H]_s[HCO_3]_s\right)\right] \qquad (84)$$

Physically dissolved oxygen:

$$\left(\frac{dP_{s_{O_2}}}{dt}\right) = \left(\frac{1}{\alpha_{O_2}V_s}\right)\left[D_{pso}\left(P_{B_{O_2}} - P_{s_{O_2}}\right) - D_{sco}\left(P_{s_{O_2}} - P_{c_{O_2}}\right)\right] \qquad (85)$$

Bicarbonate ion:

$$\left(\frac{d[HCO_3]_s}{dt}\right) = \left(\frac{1}{V_s}\right)\left[D_{csb}(r_{cs}[HCO_3]_c - [HCO_3]_s)\right.$$

$$(86)$$

$$\left. - D_{spb}(r_{sp}[HCO_3]_s - [HCO_3]_p) + V_s\left(k_u\alpha_{CO_2}P_{s_{CO_2}} - \frac{k_v}{k}[H]_s[HCO_3]_s\right)\right]$$

Hydrogen ion:

$$\left(\frac{d[H]_s}{dt}\right) = \left(\frac{1}{V_s}\right)\left[K_{csH}(t_{cs}[H]_c - [H]_s) - D_{sph}\left(\frac{[H]_s}{r_{sp}} - \frac{\alpha_{CO_2} k' P_{B_{CO_2}}}{[HCO_3]_p}\right)\right]\left[-\frac{2.303[H]_s}{\beta_s}\right] \quad (87)$$

Whole Blood

Dissolved carbon dioxide:

$$\left(\frac{dP_{B_{CO_2}}}{dt}\right) = \left(\frac{1}{\alpha_{CO_2} V_B}\right)\left[\alpha_{CO_2} Q_B\left(P_{B_{CO_2 a}} - P_{B_{CO_2}}\right) + D_{spc}\left(P_{s_{CO_2}} - P_{B_{CO_2}}\right) - V_{prpr} - V_{erer} - V_{ercarb}\right] \quad (88)$$

Oxygen: Physically dissolved and in chemical combination with hemoglobin

$$\left(\frac{dP_{B_{O_2}}}{dt}\right) = \left(\frac{1}{\bar{\alpha}_{O_2} V_B}\right)\left[Q_B\left(C_{B_{O_2 a}} - C_{B_{O_2}}\right) - D_{pso}\left(P_{B_{O_2}} - P_{s_{O_2}}\right)\right] \quad (89)$$

Blood Plasma

$$\left(\frac{d[HCO_3]_p}{dt}\right) = \left(\frac{1}{V_{B_p}}\right)\left[Q_{B_p}\left([HCO_3]_{p_a} - [HCO_3]_p\right) + D_{spb}(r_{spb}[HCO_3]_s - [HCO_3]_p) - R_{epB} + V_{prpr}\right] \quad (90)$$

where

$$V_{prpr} = (f_1/f_2)$$

$$V_{erer} = e_1 + e_2(f_1/f_2)$$

$$R_{epB} = b_8 + b_9(f_1/f_2) + b_{10}(e_1 + e_2(f_1/f_2))$$

$$f_1 = c_5 - e_1c_4$$

$$f_2 = e_2c_4 - c_6$$

$$c_4 = 1 + (1 + c_3)b_{10}$$

$$c_5 = c_2 - (1 + c_3)b_8$$

$$c_6 = c_3 - (1 + c_3)b_9$$

$$e_1 = (b_1 + b_2b_3 - rd_1 - rd_2d_3)/rd_2$$

$$e_2 = (b_3/rd_2)$$

$$d_1 = (b_1/r)$$

$$d_2 = -\left(\frac{2.303\alpha_{CO_2}k'}{2V_{B_e}\beta_e r}\right)\left(\frac{P_{B_{CO_2}a}}{[HCO_3]_{p_a}} + \frac{P_{B_{CO_2}}}{[HCO_3]_p}\right)$$

$$d_3 = 1.5V_{ercarb} + 0.6\ RO_2Hb$$

$$c_1 = \left(\frac{rQ_{B_e}}{V_{B_e}}\right)([HCO_3]_{p_a} - [HCO_3]_p)$$

$$c_2 = (rb_6 - c_1)V_{B_e}$$

$$c_3 = \left(r \frac{V_{B_e}}{V_{B_p}} \right)$$

$$b_8 = \left(\frac{V_{B_p}}{b_4} \right) (b_4 b_6 + b_5 b_7 - b_1 - b_2 b_3)$$

$$b_9 = \left(\frac{V_{B_p}}{b_4} \right) \left(\frac{b_4}{V_{B_p}} - \frac{b_5}{V_B} - b_3 \right)$$

$$b_{10} = - \left(\frac{V_{B_p}}{b_4} \right) \left(\frac{b_5}{V_B} \right)$$

$$b_1 = \left(\frac{Q_{B_p}\ {CO_2}^{k'}}{V_{B_p}} \right) \left(\frac{P_{B_{CO_2 a}}}{[HCO_3]_{p_a}} - \frac{P_{B_{CO_2}}}{[HCO_3]_p} \right)$$

$$b_2 = D_{sph} \left(\frac{[H]_s}{r_{sp}} - \frac{\alpha_{CO_2} k' P_{B_{CO_2}}}{[HCO_3]_p} \right)$$

$$b_3 = - \left(\frac{2.303\ \ {CO_2}^{k'}}{2 V_{B_p} \beta_p} \right) \left(\frac{P_{B_{CO_2 a}}}{[HCO_3]_{p_a}} + \frac{P_{B_{CO_2}}}{[HCO_3]_p} \right)$$

$$b_4 = - \frac{\alpha_{CO_2} k' P_{B_{CO_2}}}{([HCO_3]_p)^2}$$

$$b_5 = \left(\frac{k'}{[HCO_3]_p} \right)$$

$$b_6 = \left(\frac{Q_{B_p}}{V_{B_p}}\right)\left([HCO_3]_{p_a} - [HCO_3]_p\right) + \left(\frac{D_{spB}}{V_{B_p}}\right)(r_{sp}[HCO_3]_s - [HCO_3]_p)$$

$$b_7 = \left(\frac{1}{V_B}\right)\left[\alpha_{CO_2}Q_B\left(P_{B_{CO_2a}} - P_{B_{CO_2}}\right) + D_{psc}\left(P_{s_{CO_2}} - P_{B_{CO_2}}\right) - V_{ercarb}\right]$$

$$V_{ercarb} = - Q_{B_e}(carb_a - carb)$$

$$carb_a = \left(\frac{[H]_b G_a [HCO_3]_{p_a}}{a_4 + G_a[HCO_3]_{p_a}}\right)$$

$$G_a = \frac{a_{1_a}[HCO_3]_{p_a}}{\left(r[HCO_3]_{p_a}k_z + \alpha_{CO_2}k'PB_{CO_2a}\right)} + \frac{a_{2_a}[HCO_3]_{p_a}}{\left(r[HCO_3]_{p_a}k_0 + \alpha_{CO_2}k'P_{B_{CO_2a}}\right)}$$

$$a_{1_a} = (1 - s_a/100)rk_z$$

$$a_{2_a} = (s_a/100)rk_0$$

$$a_4 = (k'/rk_c)$$

$$s_a = \left(\frac{C_{B_{O_2a}} - \alpha_{O_2}P_{B_{O_2}}}{C_{max}}\right) \cdot 100$$

$$\text{carb} = \left(\frac{[H_b]G[HCO_3]_p}{a_4 + G[HCO_3]_p}\right)$$

$$G = \frac{a_1[HCO_3]_p}{\left(r[HCO_3]_p k_z + \alpha_{CO_2} k' P_{B_{CO_2}}\right)} + \frac{a_2[HCO_3]_p}{\left(r[HCO_3]_p k_0 + \alpha_{CO_2} k' P_{B_{CO_2}}\right)}$$

$$a_1 = (1 - s/100) r k_z$$

$$a_2 = (s/100) r k_0$$

$$s = \left(\frac{C_{B_{O_2}} - \alpha_{O_2} P_{B_{O_2}}}{C_{max}}\right) \cdot 100$$

$$RO_2Hb = \frac{Q_B C_{max}(s - s_a)}{(100)(22.4)}$$

$$C_{B_{O_2}} = C_{max}\left(\frac{u}{1 + u}\right) + \alpha_{O_2} P_{B_{O_2}}$$

$$u = 0.925v + 2.8v^2 + 30\ v^3$$

$$v = [0.004273 + .04326\left(P_{B_{CO_2}}\right)^{-.535}] P_{B_{O_2}}$$

$$C_{B_{O_2 a}} = C_{max}\left(\frac{u_a}{1 + u_a}\right) + \alpha_{O_2} P_{B_{O_2 a}}$$

$$u_a = 0.925v_a + 2.8v_a^2 + 30v_a^3$$

$$v_a = [0.004273 + 0.04326\left(P_{B_{CO_2 a}}\right)^{-.535}] P_{B_{O_2 a}}$$

$$\overline{\alpha}_{O_2} = \frac{C_{max}(0.925v + 2.8v^2 + 30v^3)\left(0.004273 + 0.04326\left(P_{B_{CO_2}}\right)^{-.535}\right)}{(1 - u)^2}$$

$$+ \alpha_{O_2}$$

$$\overline{\alpha}_{m_{O_2}} = \frac{\frac{C_{m_{max}}(321.221)}{100}}{\left(3.228 + P_{C_{O_2}}\right)^2} + \alpha_{O_2}$$

System Parameters and Initial Conditions: Normal Man at Rest (Steady State)

$Q_B = 0.84$ (L/min)

$\left(M_{T_{CO_2}}\right)_0 = 0.0425/22.4$ (L/min)

$\left(M_{T_{O_2}}\right)_0 = 0.05$ (L/min)

$r = 0.7$ (dimensionless)

$C_{max} = 0.201\left(\frac{LO_2}{L\ blood}\right)$

$[Hb] = 0.02058$ (M)/(L erythrocyte)

$\alpha_{CO_2} = 3 \times 10^{-5}$ (M)/(L blood)(mm Hg)

$\alpha_{O_2} = 3 \times 10^{-5}$ (M)/(L blood)(mm Hg)

$k_u = (0.13)(60)$ (1/min)

$k_v = (89)(60)$ (1/min)

$$k' = (10)^{-6.1} \text{ (M/L)}$$

$$k = k'\left(\frac{k_v}{k_u}\right) \text{ (M/L)}$$

$$k_z = 7.2 \times 10^{-8} \text{ (M/L)}$$

$$k_0 = 8.4 \times 10^{-9} \text{ (M/L)}$$

$$k_c = 2.4 \times 10^{-5} \text{ (M/L)}$$

$$HCRIT = 0.39 \text{ (dimensionless)}$$

$$\beta_p = -0.0061 \text{ M/(L)(PH unit)}$$

$$\beta_e = -0.056 \text{ M/(L)(PH unit)}$$

$$\beta_s = -0.0021 \text{ M/(L)(PH unit)}$$

$$\beta_c = -0.0195 \text{ M/(L)(PH unit)}$$

$$V_B = 1.0 \text{ (L)}$$

$$V_s = 3.46 \text{ (L)}$$

$$V_c = 26.5 \text{ (L)}$$

$$Q_{B_p} = Q_B(1 - HCRIT/100) \text{ (L/min)}$$

$$Q_{B_e} = Q_B(HCRIT/100) \text{ (L/min)}$$

$$V_{B_p} = V_B(1 - HCRIT/100) \text{ (L/min)}$$

$$V_{B_e} = V_B(HCRIT/100) \text{ (L/min)}$$

$$C_{m_{max}} = 1.64 \times 10^{-4} \text{ (L oxymyoglobin/L ICF)}$$

$$M_{T_{O_2}} = \left(M_{T_{O_2}}\right)_0 \qquad P_{c_{O_2}} \geq 10.0$$

$$= \left(M_{T_{O_2}}\right)_0 \left(\frac{P_{c_{O_2}}}{10}\right) \qquad P_{c_{O_2}} < 10.0$$

$$M_{T_{CO_2}} = \left(M_{T_{CO_2}}\right)_0 \qquad P_{c_{O_2}} \geq 10.0$$

$$= \frac{\left(M_{T_{CO_2}}\right)_0}{\left(M_{T_{O_2}}\right)_0} \left(M_{T_{O_2}}\right) \qquad P_{c_{O_2}} < 10.0$$

$$P_{B_{CO_2 a}} = 39.0 \text{ (mm Hg)}$$

$$P_{B_{O_2 a}} = 100.0 \text{ (mm Hg)}$$

$$[HCO_3]_{p_a} = 0.022888 \text{ (M/L)}$$

$P_{B_{CO_2}}$, $P_{B_{O_2}}$, $[HCO_3]_p$: Solution to the following three non-linear algebraic equations.

$$\alpha_{CO_2} Q_B \left(P_{B_{CO_2 a}} - P_{B_{CO_2}}\right) + \left(Q_{B_p} + rQ_{B_e}\right)\left([HCO_3]_{p_a} - [HCO_3]_p\right)$$

$$+ M_{T_{CO_2}} + Q_{B_e}(\text{carb}_a - \text{carb}) = 0$$

$$\left(Q_{B_p} + rQ_{B_e}\right)\left([HCO_3]_{p_a} - [HCO_3]_p\right) + 2\left(\frac{Q_{B_p}\beta_p + Q_{B_e}\beta_e}{2.303}\right) \cdot$$

$$\left(\frac{\dfrac{P_{B_{CO_2a}}}{[HCO_3]_{p_a}} - \dfrac{P_{B_{CO_2}}}{[HCO_3]_p}}{\dfrac{P_{B_{CO_2a}}}{[HCO_3]_{p_a}} + \dfrac{P_{B_{CO_2}}}{[HCO_3]_p}}\right) + 1.5Q_{B_e}(carb_a - carb) - 0.6RO_2Hb = 0$$

$$C_{B_{O_2a}} - \left(\frac{M_{T_{O_2}}}{Q_B}\right) - C_{max}\left(\frac{u}{1 + u}\right) - \alpha_{O_2}P_{B_{O_2}} = 0$$

$(\Delta P)_{s_{B_{CO_2}}} = 2.0$ mm Hg

$(\Delta P)_{s_{B_{O_2}}} = 5.0$ mm Hg

$(\Delta P)_{s_{c_{CO_2}}} = 5.0$ mm Hg

$(\Delta P)_{s_{c_{O_2}}} = 5.0$ mm Hg

$x = 0.90$ (dimensionless)

$z = 0.10$ (dimensionless)

$F = 0.80$ (dimensionless)

$t_{cs} = 0.5$ (dimensionless)

$r_{es} = 29.88$ (dimensionless)

$$r_{sp} = 0.95 \text{ (dimensionless)}$$

$$P_{s_{CO_2}} = P_{B_{CO_2}} + (\Delta P)_{s_{B_{CO_2}}}$$

$$[HCO_3]_s = 0.027 \text{ (M/L)}$$

$$D_{spc} = \frac{\left(M_{T_{CO_2}}\right) \cdot (x - z)}{(\Delta P)_{s_{B_{CO_2}}}}$$

$$[H]_s = \frac{k_u \alpha_{CO_2} P_{s_{CO_2}} - M_{T_{CO_2}}\left(\frac{z}{\bar{V}_s}\right)}{\frac{k_v}{k}[HCO_3]_p}$$

$$D_{spB} = \frac{\left(M_{T_{CO_2}}\right) \cdot (1 + z - x)}{(r_{sp}[HCO_3]_s - [HCO_3]_p)}$$

$$D_{spH} = \frac{\left(M_{T_{CO_2}}\right) \cdot (1 + z - x)}{\left(\frac{[H]_s}{r_{sp}} - \frac{\alpha_{CO_2} k' P_{B_{CO_2}}}{[HCO_3]_p}\right)}$$

$$P_{c_{CO_2}} = P_{s_{CO_2}} + (\Delta P)_{s_{c_{CO_2}}}$$

$$[HCO_3]_c = 0.012 \text{ (M/L)}$$

$$D_{csc} = \frac{\left(M_{T_{CO_2}}\right) \cdot x}{(\Delta P)_{s_{c_{CO_2}}}}$$

$$[H]_c = \frac{\left(k_u \alpha_{CO_2} P_{c_{CO_2}} - \left(M_{T_{CO_2}}\right) \frac{(F - x)}{V_c}\right)}{\left(\frac{k_v}{k}\right) [HCO_3]_c}$$

$$D_{csB} = \frac{\left(M_{T_{CO_2}}\right) \cdot (1 - x)}{\left(r_{cs} [HCO_3]_c - [HCO_3]_s\right)}$$

$$K_{csH} = \frac{\left(M_{T_{CO_2}}\right) \cdot (1 - x)}{\left(t_{cs} [H]_c - [H]_s\right)}$$

$$P_{s_{O_2}} = P_{B_{O_2}} - (\Delta P)_{s_{B_{O_2}}}$$

$$P_{c_{O_2}} = P_{s_{O_2}} - (\Delta P)_{s_{c_{O_2}}}$$

$$D_{pso} = \frac{\left(M_{T_{O_2}}\right)}{(\Delta P)_{s_{B_{O_2}}}}$$

$$D_{sco} = \frac{\left(M_{T_{O_2}}\right)}{(\Delta P)_{s_{c_{O_2}}}}$$

Figures 3.7.1 through 3.7.4 show the transient response of the eleven variables of this model, while in Table 3.7 the performance of the algorithms can be observed.

For error allowance of approximately 10^{-7} the three algorithms of GEARSB had convergence problems, however at error allowance of $\simeq 10^{-5}$, the three algorithms of such a program were more efficient than all of the others. The superiority of the option 1 for stiff systems algorithm ranged from 1:8 if compared with Polynomial Extrapolation of DIFSUB, tc 1:15 if compared with DESUB.

Because this is an eleven equation system, the overhead of numerically approximating the terms of the Jacobian was significant, and caused a difference in efficiency of approximately 6.6% between the two options for stiff systems in GEARSB.

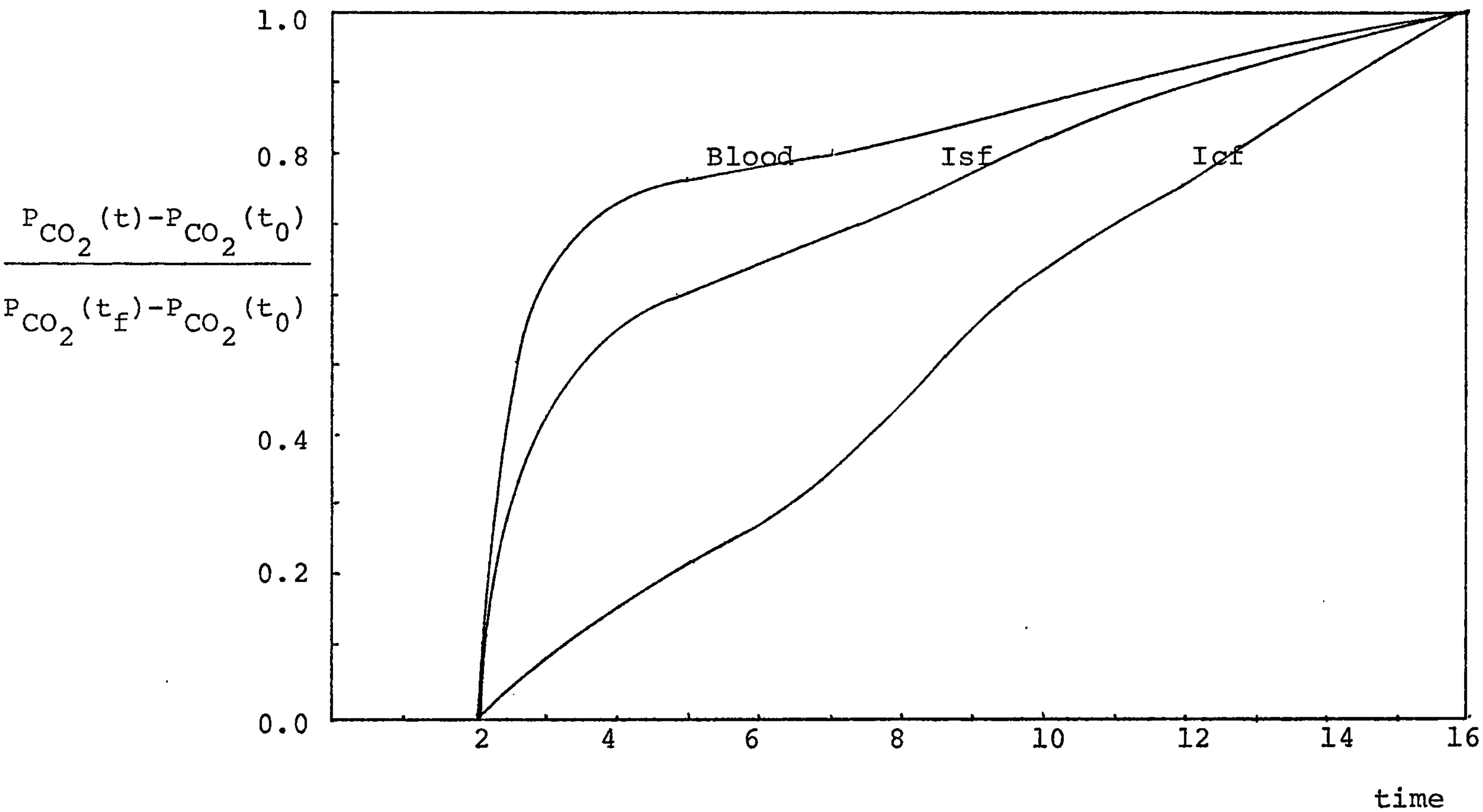


Fig. 3.7.1 Mathematical Model of Human muscle.

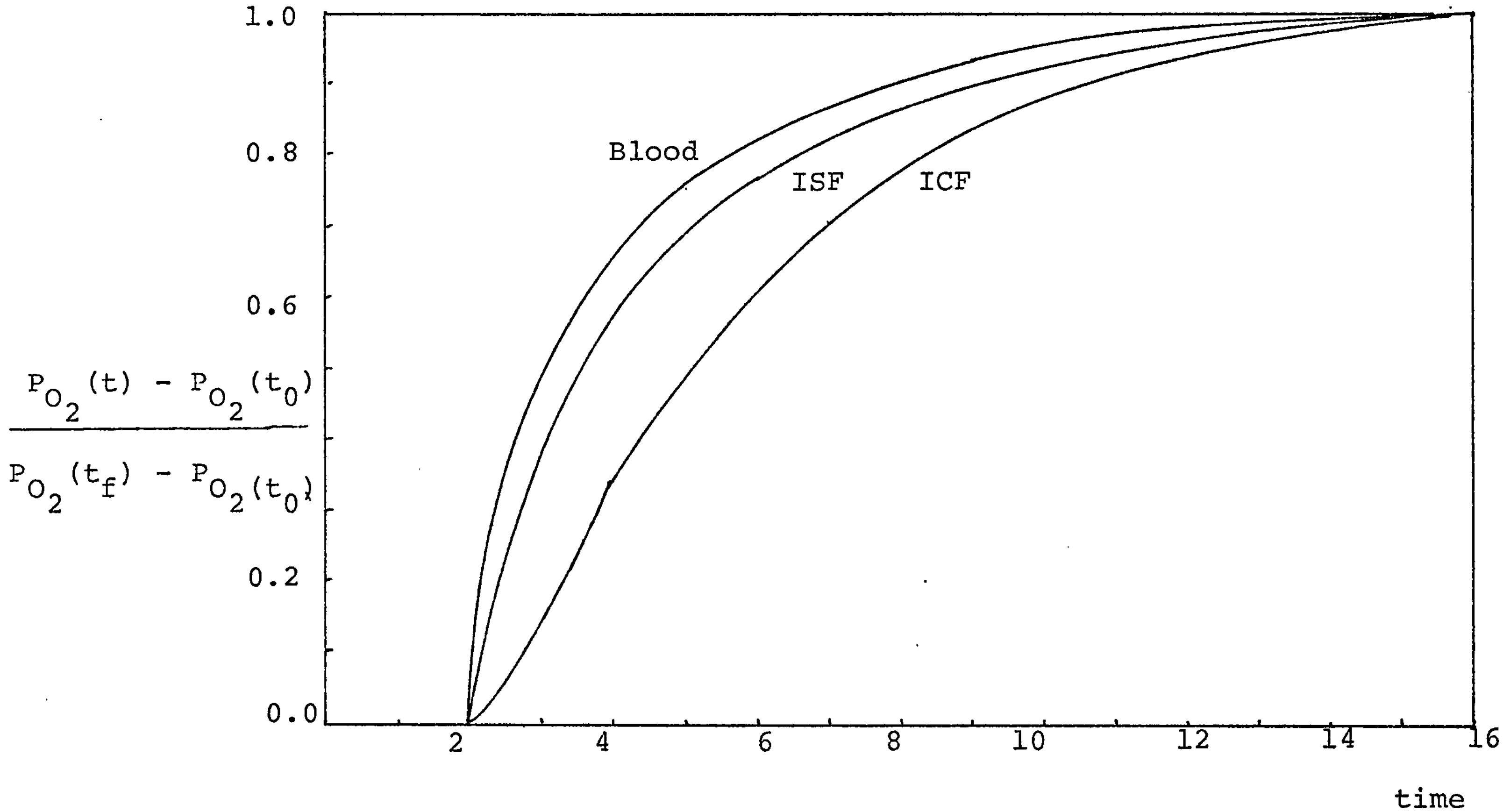


Fig. 3.7.2 Mathematical Model of Human Muscle.

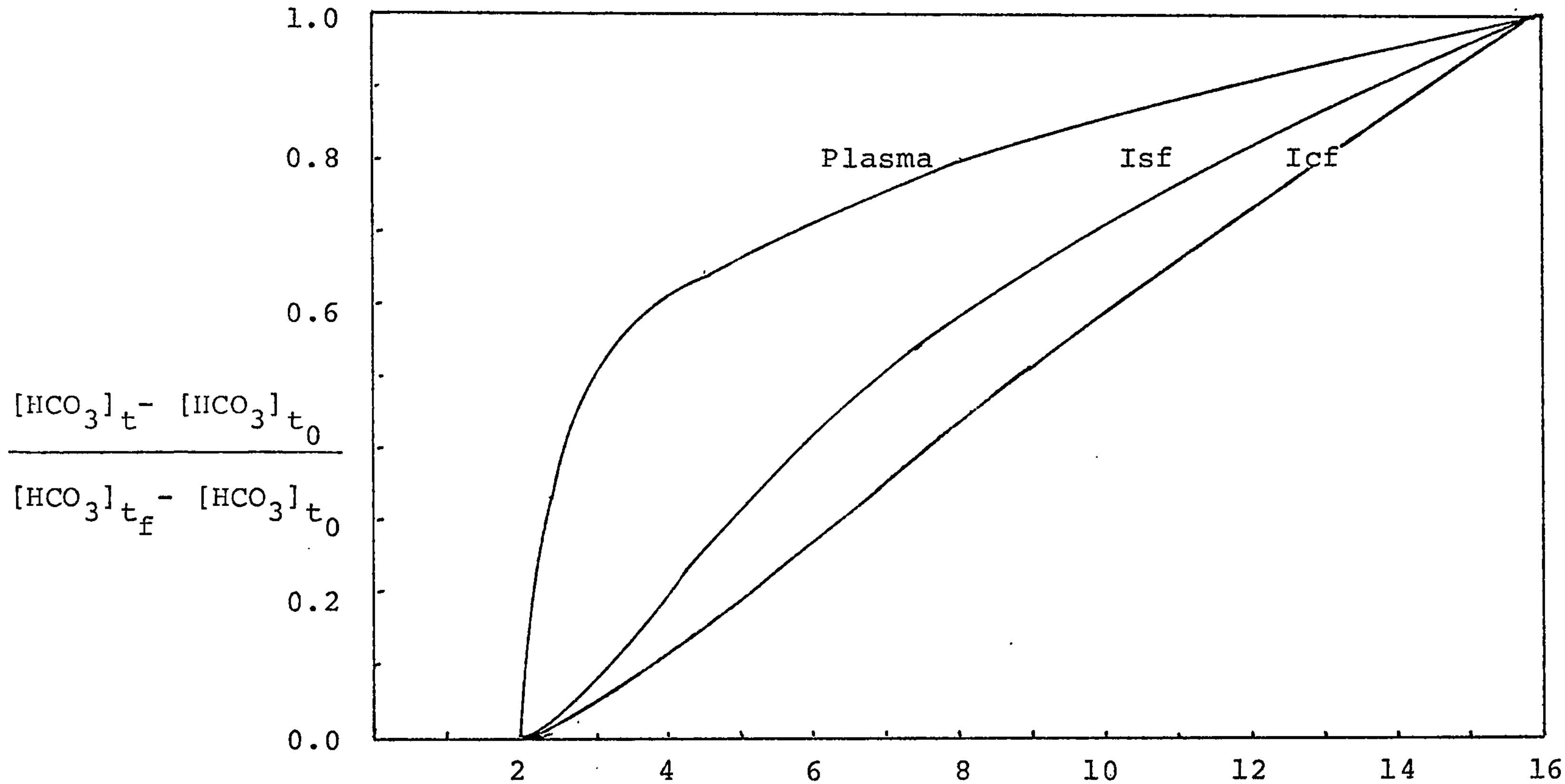


Fig. 3.7.3 Mathematical Model of Human Muscle.

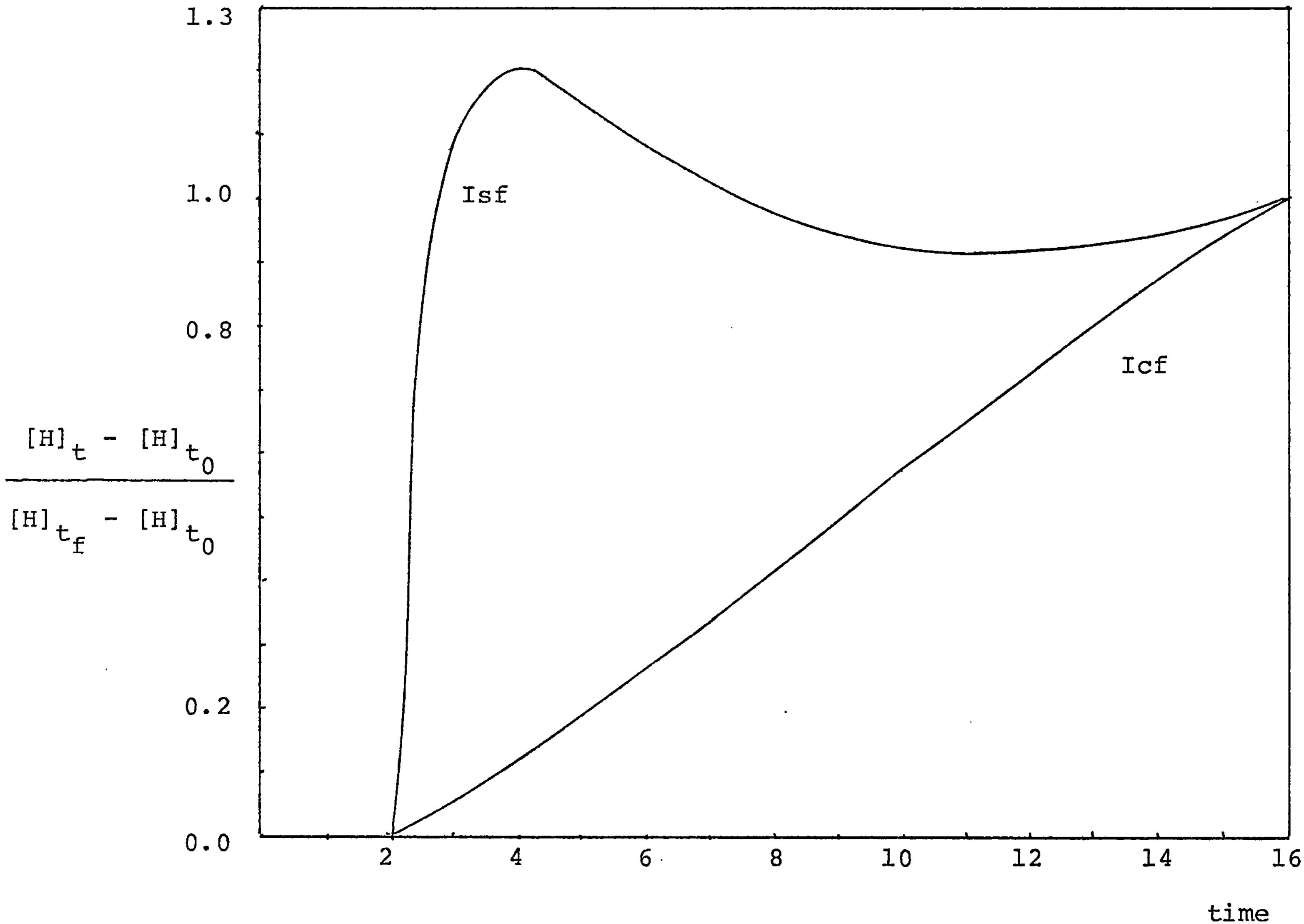


Fig. 3.7.4 Mathematical Model of Human Muscle.

		Error = 1×10^{-3}		1×10^{-5}			1×10^{-7}	
Program	Algorithm	Steps	Func. Eval.	Steps	Func. Eval.	Time Secs.	Steps	Func. Eval.
DIFSUB	Rational Extrapolation	713	19635	717	19821	630.86	736	20418
	Polynomial Extrapolation	704	18735	709	19059	588.65	723	19797
DESUB	Double Precision		28920		29228	1073		31037
	Single Precision							
GEARSB	Adam's Predictor-Corrector	2468	6477	2710	7353	268.2	b	
	Gear's Stiff Option 1	969	1014	984	1130	72	b	
	Gear's Stiff Option 2	1063	3326	987	1311	76.76	b	
Runge-Kutta	Runge-Kutta-Gill	1600	19200	1600	19200	614.81	1610	19312

Table 3.7 A Biological System

b no convergence

3.8 Discussion of Results

For the purpose of discussion of results, it is convenient to separate the problems as follows:

I. Moderately stiff systems

1) A stirred tank reactor with exothermic reaction.

2) A turbulent boundary layer on a flat plate.

3) A periodic chemical reaction sinusiodal forcing function.

II. Highly stiff systems

1) A system of reaction rate equations.

2) A periodic chemical reaction bang-bang forcing function.

3) The thermal decomposition of ozone.

4) The transient behavior of a catalytic fluidized bed.

5) Mathematical model of muscle chemical aspects.

It will also be convenient to identify the different algorithms as follows:

A-1. Bulirsch and Stoer Rational Extrapolation in DIFSUB.

A-2. Polynomial Extrapolation in DIFSUB.

A-3. Double Precision version of DESUB

A-4. Single Precision version of DESUB

A-5. Adam's Predictor Corrector in Gear program.

A-6. Stiff algorithm Option 1 in Gear program.

A-7. Stiff algorithm Option 2 in Gear program.

A-8. Runge-Kutta-Merson.

In this section the error allowance criterion will be indicated as EPS.

3.8.1 Comparison of the Overall Efficiency of the Algorithms

In order to compare the overall efficiency of the different methods, the integration times of the problems were normalized (division of the computational times of the different methods by the time of the best for every example) and are shown in Tables 3.8.1.A and 3.8.1.B (Algorithm A-4 was not considered.)

From these tables it can be seen that for the stiff systems, algorithm A-6 was the most efficient in all cases. The difference in efficiency between this algorithm and the algorithms in the rest of the programs ranged from 1:1.7 in problem II-3 solved by algorithm A-2, to 1:952 in I-1 solved by A-3.

For the moderate systems the three algorithms of Gear's program are superior to the others, however this superiority is not as notable as for the stiff systems.

Problem	Runge-Kutta-Merson	DESUB	DIFSUB		GEARSB		
		Double Precision	Rational Extrap.	Polynomial Extrap.	Adam's P. - C.	Stiff Option 1	Stiff Option 2
3.1 A System of Reaction Rate Equations	343.219	951.658	437.350	a	467.675	1	1.044
3.6 The Transient Behavior of a Catalytic Fluidized Bed	123.051	a	125.352	99.404	119.087	1	1.038
3.7 A Biological System	8.539*	14.903	8.762	8.176	3.725	1	1.066
3.5 The Thermal Decomposition of Ozone	2.321	3.318	2.203	1.712	2.884	1	1.014
3.4.B Periodic Chemical Reaction Bang-bang Control	8.524	a	4.774	4.258	3.596	1	1.036

Table 3.8.1.A Normalized Times for Stiff Systems

a unstable
*Runge-Kutta-Gill

Problem	Runge-Kutta-Merson	DESUB	DIFSUB		GEARSB		
		Double Precision	Rational Extrap.	Polynomial Extrap.	Adam's P. - C.	Stiff Option 1	Stiff Option 2
3.2 The Stirred Tank Reactor	1.083	1.341	1.195	1.052	1.233	1	1
3.3 The Turbulent Boundary Layer	1.250	1.784	1.526	1.387	1	1.063	1.160
3.4.A Periodic Chemical Reaction Sinusoidal Control	3.834	1.241	1.971	1.532	1.052	1	1.014

Table 3.8.1.B Normalized Times for Moderate Systems

3.8.2 Comparison of the Effect of the Error Allowance on the Different Methods

Table 3.8.1.A shows that for EPS ≃ 10^{-5} algorithm A-6 was superior to all other algorithms by a factor ranging from 1:8 to 1:14 in problem II-5, and by a factor of 1:4 to 1:8 in II-2. However at EPS ≃ 10^{-7} algorithms A-5, A-6 and A-7 had convergence problems, while the other algorithms lacked those problems. Thus it seems that for systems like those mentioned, the price to be paid for a very efficient solution is to allow EPS ≃ 10^{-5} or larger. For EPS ≃ 10^{-7} a good choice should be algorithm A-2.

Algorithm A-4 consistently had convergence problems or was found to be very inefficient for EPS ≃ 10^{-7}. More detailed information about this point is provided in section 3.8.4.

3.8.3 Comparison of Stability of the Different Algorithms

Algorithms A-6 and A-7 were stable in all the runs attempted. Algorithm A-5 was unstable only in the solution of II-1 for EPS ≃ 10^{-3}, but it was stable for EPS $\gtrsim 10^{-4}$. A-1 was unstable also for II-1 for EPS ≃ 10^{-3}. The rest of the algorithms showed unstability in at least two cases.

3.8.4 Comparison of Single and Double Precision Versions of DESUB

In general algorithm A-4 works reasonably well for EPS ≃ 10^{-5}, however as a higher accuracy is required, the efficiency of this algorithm decreases and at EPS ≃ 10^{-7}, its performance is really bad (too many function evaluations) or there is no convergence.

To illustrate these points the following facts can be mentioned:

a) In solving I-1 for an EPS $\approx 10^{-5}$, A-4 took 1.88 seconds (a-3 took 2.86), while in attempting to solve the same problem for an EPS $\approx 10^{-7}$ with no printed output it took more than forty-two minutes.

b) Table 3.1 shows that in the solution of the problem II-1 for EPS $\approx 10^{-5}$ the single precision version took 93% of the time taken by the algorithm A-3 while at EPS $\approx 10^{-7}$ the double precision version required 361210 function evaluations to perform the whole integration and the single precision version took 87191 for only 10% of the integration.

c) In II-3, at EPS $\approx 10^{-5}$ it can be seen that the behavior of both versions of DESUB is very similar. However for an EPS $\approx 10^{-7}$, A-4 took more function evaluation than A-3 by a factor of 3.574.

d) In I-2, while algorithm A-3 solved the problem at EPS $\approx 10^{-7}$, algorithm A-4 lacked convergence for such an EPS.

3.8.5 Comparison of the two Algorithms for Stiff Systems in GEARSB Program

The difference in integration time among algorithms A-6 and A-7 is due to the fact that while A-6 needs only one evaluation of the analytically expressed terms of the Jacobian every time that this matrix has to be evaluated, algorithm A-7 requires a number of evaluations equal to the number of dependent variables of the right hand side of the differential equations to numerically approximate such a matrix. As can be seen this difference can be significant for large systems. In problem II-5, built by eleven equations, the difference in function evaluations between those algorithms was 181 while in computational time the difference was ≃ 6.6%.

It is also interesting to note that in the same problem (II-5) algorithm A-7 took 3326 function evaluations for EPS ≃ 10^{-3}, while the same algorithm took 1311 for EPS ≃ 10^{-5}. This can be explained considering that in the numerical approximation of the Jacobian, the change (R) made to each dependent variable to disturb the system (and so evaluate the corresponding term of the Jacobian) is calculated as a direct function of EPS, therefore for large EPS's, R is also large. In some cases this can originate inaccuracies in the numerical approximation that will slow the convergence. This seems to be the case in this problem since for EPS ≃ 10^{-4} the same algorithm took 1612 function evaluations.

3.8.6 Comparison of the Extrapolation Algorithms

In general A-1 and A-2 had better performance than A-3. This can be better seen in the highly stiff systems. If A-1 and A-2 are compared, A-2 is more efficient that A-1 by an average factor of approximately 1.20:1. However, in problem II-1, A-2 was unstable at EPS ≃ 10^{-3} and at EPS ≃ 10^{-5}, while A-1 was unstable only at EPS ≃ 10^{-3}.

3.8.7 Performance of Runge-Kutta-Merson Algorithm

For stiff systems algorithm A-8 is very inefficient. Such an inefficiency can be seen in Table 3.8.1 which shows that its best performance in problem II-3 is 2.3 times slower than the best algorithm for such a problem, while in problem II-1 the factor was 343. For moderate systems its performance is acceptable.

Chapter 4

CONCLUSIONS

From the results obtained in Chapter 3, it can be seen that the three algorithms in the program written by Gear can, if properly selected, solve efficiently stiff as well as non-stiff systems of ordinary differential equations. The Adam's methods whose extraneous eigenvalues are zero is usually the best choice to solve any problem [2]. Therefore such an algorithm should be attempted first. If it is found that the problem is stiff, any of the two options for stiff systems should be used.

To increase the accessibility of Gear programs for general engineering usage, a driver set of subroutines (DRGERT, DATARD, INVAR, PARESC and ESCINT) was written with the following characteristics:

a) Minimization of input requirements. The user has to specify only the following parameters

1) The number of equations
2) The printed output interval
3) The required error allowance
4) Initial value of both dependent and independent variables
5) Final value of the independent variable
6) Algorithm to be used (optional).

The rest of the parameters to the Gear program are properly handled by the driver set of subroutines.

b) The driver system increases automatically the error allowance as required up to three orders of magnitude to avoid convergence problems. (This in some cases prevents the premature stoppage of the integration because of the fact that the requested error is smaller than can be handled.) .

c) The driver system has flexibility to allow the user to provide his own output subroutine, and it also provides a standard printed output for all of the users who do not have special output requirements.

d) The initial values of the dependent variables can be provided through punched card or the user can provide a subroutine to calculate them. The subroutine (DATARD) has card reading instructions for such variables that will be bypassed if the user inserts his own initial values subroutine.

e) Because Gear's method automatically adjusts the step size (and order) as the calculation proceeds to achieve specific accuracy requirements, the solution at specific values of the independent variable is not available. Considering that in some cases this is an important factor, an algorithm was implemented in the subroutine (ESCINT) to obtain the solution at such specified points. This feature should be used rationally since it implies a reduction of efficiency.

Thus the user can select between:

1) Asking for the solution at specific values of the independent variable.

2) Asking for the solution at values between certain points of the independent variable. ($(t_{n+1} - t_n) \geq t_p$, where t_p is a constant selected by the user.)

f) Because the program has three algorithms that can solve efficiently most of the problems, it is important to dynamically select the adequate algorithm for the most efficient solution of the problem in question.

To handle this situation, the driver system offers three alternatives

1) The user decides which algorithm to use. For the cases in which the user already knows the characteristics of the problem that is being solved.

2) The user lets the driver system decide between using Adam's Predictor Corrector or the Option 1 for stiff systems. In this case, the user has to include a subroutine (PEDERV) to calculate the terms of the Jacobian.

3) The user lets the driver system decide between using Adam's Predictor Corrector or the Option 2 for stiff systems. In this case the Jacobian is numerically approximated if the algorithm selected is the stiff Option 2.

For the selection mentioned in points 2 and 3, Schacham [private communication] proposed the following strategy:

The integration is started with the Adam's method, using an error allowance EPS_1. The integration in continued until a stable value for step size h_1 is obtained. The integration of the same interval is repeated using another error allowance EPS_2, where $EPS_2 >> EPS_1$. If the step size is limited by the accuracy requirements, then the new step size will be

$$h_2 = h_1 \left(\frac{EPS_2}{EPS_1} \right)^{\frac{1}{n}}$$

where n is the order of the integration method.

If this expression holds, then the integration may be continued with the Adam's method. If the new step size is much smaller than expected or $h_2 \approx h_1$, then the step size is limited by stability, and the integration must be continued using the stiff algorithm.

This strategy was implemented using the average value of the step size rather than its last value, and for the cases in which the order was different (in the two integrations), the following expression was used.

$$h_2 = h_1 \left(\frac{EPS_2^{\frac{1}{n_2}}}{EPS_1^{\frac{1}{n_1}}} \right)$$

This selection algorithm had satisfactory performance in all the problems attempted as can be seen in Appendix A.

Listings of all the algorithms used as well as example problems can be found in the appendix section.

Many problems in continuous system simulation lead to systems of ordinary differential equations which require the solution of a simultaneous set of non-linear and implicit algebraic equations each time that the derivatives are to be evaluated. Thus, the actual system should be extended to provide the capability of simultaneous numerical solution of differential-algebraic equations. A unified method for handling this is discussed by Gear [22], and would be a friutful extension of this work.

NOMENCLATURE

y'	$= dy/dt$
f	= any function
t	= time (independent variable)
y	= dependent variable
y_n	= approximation to the dependent variable at point n
$y_{n,(0)}$	= predictor approximation to y_n by linear extrapolation
B	= matrix of constants used in the predictor process
$G(y_{n,(0)})$	= amount by which $y_{n,(0)}$ does not satisfy the differential equation in the corrector process
M	= number of corrector iterations
$0(h)$	= any function of h such that there exists constants h_0 and k independent of h for which $\lvert 0(h) \rvert \leq kh$ for all $\lvert h \rvert < h_0$
C	= constant coefficient
$(y)_i$	= ith component of the vector y
$y^{(i)}$	$= d^i y/dt^i$ (ith derivative of y with respect to "t")
$R_m^i(h)$	= rational approximation which agrees with y(x,h) at $h = h_i, h_{i+1}, \ldots, h_{i+m}$ where $h_i > h_{i+1} > \ldots > h_{i+m}$
D_m^i	$= R_m^i - R_{m-1}^{i+1}$
e_m^i	$= R_m^i - R_{m-1}^i$
W_m^i	$= R_m^i - R_m^{i-1}$
$R_m^i(t,h)$	= polynomial of degree m in h

h	=	step size
$T(h)$	=	a discrete approximation for an ordinary differential equation
δ_j	=	difference between two sucessive extrapolated values for the jth component
EP	=	error tolerance
γ_j	=	$(-1)^j \int_0^1 \binom{-s}{j} ds$
β_{ki}	=	$(-1)^{i-1} \sum_{j=i-1}^{k-1} \gamma_j \binom{j}{i-1}$
γ_j^*	=	$(-1)^1 \int_0^1 \binom{-s+1}{j} ds$
β_{ki}^*	=	$(-1)^j \sum_{j=1}^{k-1} \binom{j}{i} \gamma_j^*$
ω	=	weight component
∇a_q	=	backwards difference of the last component of a
a	=	$[y, hy^1, \ldots, h^q y^{(q)}/q!]^T$ (T = transpose operator)
$\lVert \ \rVert_2$	=	L_2 norm
ε	=	error allowed
λ	=	eigenvalue
$\sigma(\xi)$	=	$\sum_{i=0}^{k} \beta_i \xi^{k-1}$
$\rho(\xi)$	=	$\sum_{i=0}^{k} \alpha_i \xi^{k-1}$
μ	=	$h\lambda$ plane
$\bar{x}$	=	vector x
$\bar{\bar{x}}$	=	matrix x
s	=	$(t - t_{n-1})/h$
L_0, L_1	=	compenents of vector I

BIBLIOGRAPHY

1. Gear, C. W., "The Automatic Integration of Stiff Ordinary Differential Equations," Information Processing, ed. A. J. H. Morrel. Amsterdam, North Holland: North Holland Publishing Company, 1969.

2. Gear, C. W. Numerical Initial Value Problems in Ordinary Differential Equations. Englewood Cliffs, New Jersey: Prentice Hall, 1971.

3. Sienfeld, J., L. Lapidus and H. Hwang, "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations," Industrial and Engineering Chemistry Fundamentals, IX, No. 2 (1970), 266.

4. Hull, T. E., W. H. Enright, B. M. Fellen and A. E. Sedwick, "Comparing Numerical Methods for Ordinary Differential Equations," SIAM Journal of Numerical Analysis, IX, No. 4 (December, 1972), 603.

5. Distefano, G. P., "Stability of Numerical Integration Techniques," AIChE Journal, XIV, No. 6 (November, 1968), 946.

6. Distefano, G. P., "Mathematical Modeling and Numerical Integration of Multicomponent Batch Distillation Equations," AIChE Journal, XIV, No. 1 (January, 1968), 190.

7. Fox, P. A., "DESUB: Integration of a First Order System of Ordinary Differential Equations," Mathematical Software, ed. J. R. Rice. New York: Academic Press, 1971.

8. Merson, R. H., "An Operational Method for the Study of Integration Processes," Proceedings of a Symposium on Data Processing. Salisbury, South Australia: Weapons Research Establishment, 1957.

9. Bulirsch, R. and J. Stoer, "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods," Numerical Mathematics, VIII (1966), 1-13.

10. Clark, N. W. "Program Description for Library Subroutine." Report No. ANL D250-DIFSUB of Argonne National Laboratories, Argonne, Illinois, 1966.

11. Ince, E. L. Ordinary Differential Equations. Dover, New York: Dover Publications, 1956.

12. Lowell, David K., FSTIME (IBM/360 Assembly Language) Subroutine, Engineering Systems Simulation Laboratory User's Guide, University of Houston.

13. Shannon, P. "Numerical Integration of Stiff, Sensitive and Multivalued Equations." Unpublished M.Sc. Thesis, University of Houston, 1971.

14. Bullin, J. "Two Dimensional Turbulent Boundary Layer with Favorable Pressure Gradients: Similarity Type Solution of the Momentum Equation." Unpublished M.Sc. Thesis, University of Houston, 1970.

15. Aiken, R. C. and L. Lapidus, "An Effective Numerical Integration Method for Typical Stiff Systems," AIChE Journal, XX, No. 2 (March, 1974), 368.

16. Bowen, J. R., A. Acrivos and A. K. Oppenheim, "Singular Perturbation Refinement to Quasi Steady State Approximation in Chemical Kinetics," Chemical Engineering Science, XVII (1963), 172.

17. Luss, D. and N. R. Amundson, "Stability of Batch Catalytic Fluidized Beds," AIChE Journal, XIV, No. 2 (1968), 211.

18. Sincic, Dinko "Dynamic Behavior of the Forced Periodic Chemical Reactors" Unpublished M.Sc. Thesis, University of Houston, 1975.

19. Robertson, H. H., "Solution of a Set of Reaction Rate Equations," Numerical Analysis, ed. J. Walsh. Washington, D. C.: Thompson Book Company, 1967.

20. Aris, R. and N. Amundson, "An Analysis of Chemical Reactor Stability and Control," Chemical Engineering Science, VII (1958), 121.

21. Bidani, A. "Mathematical Modeling of the Human Respiratory System." Unpublished Ph.D. Dissertation, University of Houston, 1975. (Dr. R Flumerfelt Faculty Advisor).

22. Gear, C. W., "Simultaneous Numerical Solution of Differential-Algebraic Equations," IEEE Transactions, Circuit Theory, CT-18, No. 1 (January 1971), 89.

APPENDIX A

This appendix contains the following material:

1) Listing of Gear program. Identified as GEARSB through all this work, such a program was modified to allow the user to choose his own names for the subroutines to evaluate the right hand side of the differential equations, and to evaluate the terms of the Jacobian matrix. In the GEARSB version such subroutines must be named as DIFFUN and PEDERV respectively.

2) IBM Scientific Subroutine Package matrix inversion routine MINV.

3) Driver set of subroutines for GEARMF as follows:

DRGERT (Main routine).

DATARD (Data and parameter reading).

PARESC (Writing of parameters routine).

INVAR (Initialization of variables).

ESCINT (Integration driver).

4) Solution of the selected problems using different options of the driver set are presented. The solution of "A system of reaction rate equations" is presented in two forms:

a) "Natural" solution of Gear method.

b) Solution forced at exact intervals (of unity) of the independent variable.

Note the difference in function evaluations as well as integration steps taken in each case.

5) Solution of the "Mathematical model of human muscle-chemical aspects" problem.

If not explicitely presented, the main program for all the examples in part 4 is similar to the first one presented.

```
      SUBROUTINE GEARMF(N,T,Y,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG, GEAR  10
     1                   JSTART,MAXDER,PW,PPW,LLL,MMM,DIFFUN,PEDERV)    GEAR  20
      IMPLICIT REAL*8 (A-H,O-Z)                                          GEAR  30
      EXTERNAL DIFFUN,PEDERV                                             GEAR  40
C*********************************************************************** GEAR  50
C*    THIS SUBROUTINE INTEGRATES A SET OF N ORDINARY DIFFERENTIAL FIRS* GEAR  60
C*    ORDER EQUATIONS OVER ONE STEP OF LENGTH H AT EACH CALL. H CAN BE* GEAR  70
C*    SPECIFIED BY THE USER FOR EACH STEP, BUT IT MAY BE INCREASED OR * GEAR  80
C*    DECREASED BY DIFSUB WITHIN THE RANGE HMIN TO HMAX IN ORDER TO   * GEAR  90
C*    ACHIEVE AS LARGE A STEP AS POSSIBLE WHILE NOT COMMITTING A SINGLE GEAR 100
C*    STEP ERROR WHICH IS LARGER THAN EPS IN THE L-2 NORM, WHERE EACH * GEAR 110
C*    COMPONENT OF THE ERROR IS DIVIDED BY THE COMPONENTS OF YMAX.    * GEAR 120
C*                                                                    * GEAR 130
C*    THEPROGRAM REQUIRES THREE SUBROUTINES NAMED                     * GEAR 140
C--------------------------------------------------------------------- GEAR 150
C*    NOTE:                                                           * GEAR 160
C*              THIS VERSION WAS MODIFIED SO THAT IT CALLS 'MINV'     * GEAR 170
C*              FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE.             GEAR 180
C--------------------------------------------------------------------- GEAR 190
C*       DIFFUN(T,Y,DY)                                               * GEAR 200
C*       MATINV(PW,N,M,J)                                             * GEAR 210
C*       PEDERV(T,Y,PW,M)                                             * GEAR 220
C*    THE FIRST, DIFFUN, EVALUATES THE DERIVATIVES OF THE DEPENDENT   * GEAR 230
C*    VARIABLES STORED IN Y(1,I) FOR I=1 TO N, AND STORES THE         * GEAR 240
C*    DERIVATIVES IN THE ARRAY DY.  THE SECOND IS CALLED ONLY IF THE  * GEAR 250
C*    METHOD FLAG MF IS SET TO 1 OR 2 FOR STIFF METHODS. IT MUST INVERT GEAR 260
C*    THE N BY MATRIX STORED IN THE ARRAY PW(M,M). IF THE INVERSION IS* GEAR 270
C*    SUCCESFUL, J SHOULD BE SET TO 1, OTHERWISE IT SHOULD BE SET TO-1  GEAR 280
C*    PEDERV IS USED ONLY IF MF IS 1, AND COMPUTES THE PARTIAL        * GEAR 290
C*    DERIVATIVES OF THE DIFFERENTIAL EQUATIONS AS DESCRIBED UNDER THE* GEAR 300
C*    MF PARAMETER.                                                   * GEAR 310
C*                                                                    * GEAR 320
C*    THE PROGRAM USES DOUBLE PRECISION ARITHMETIC FOR ALL FLOATING   * GEAR 330
C*    POINT VARIABLES EXCEPT THOSE STARTING WITH P. THE FORMER ARE    * GEAR 340
C*    SINGLE PRECISION TO SAVE TIME AND SPACE.                        * GEAR 350
```

```
      C*                                                                 * GEAR 360
      C*    THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER INT THE    * GEAR 370
      C*    SINGLE PRECISION ARRAY PW AND THE DOUBLE PRECISION ARRAY SAVE.  * GEAR 380
      C*    THE ARRAY PW IS USED ONLY TO HOLD THE MATRIX OF THE SAME NAME,BUT GEAR 390
            CONTINUE                                                          GEAR 400
      C*    SAVE IS USED TO HOLD SEVERAL ARRAYS. THE REGIONS USED ARE        * GEAR 410
      C*      SAVE(J,I)  1.LE.J.LE.8 AND 1.LE.I.LE.N IS USED TO SAVE THE     * GEAR 420
      C*          VALUES OF Y IN CASE A STEP HAS TO BE REPEATED.             * GEAR 430
      C*      SAVE(9,I)  IS USED MAINLY TO HOLD THE CORRECTION TERMS IN THE * GEAR 440
      C*          CORRECTOR LOOP.                                            * GEAR 450
      C*    SAVE(10,I) IS USED TO SAVE THE VALUES OF THE SUMS OF ALL OF THE * GEAR 460
      C*        CORRECTION TERMS IN THE PREVIOUS STEP AFTER THEY             * GEAR 470
      C*        HAVE BEEN ACCUMULATED IN THE ARRAY ERROR INT THE             * GEAR 480
      C*        CURRENT STEP. THIS ENABLES THE BACKWARDS DIFFERENCE          * GEAR 490
      C*        OF ERROR TO BE FORMED. IT IS USED TO ESTIMATE THE            * GEAR 500
      C*        STEP SIZE FOR ONE ORDER HIGHER THAN CURRENT.                 * GEAR 510
      C*    SAVE(N1+I,1) IS USED TO STORE THE DERIVATIVES WHEN THEY ARE      * GEAR 520
      C*        COMPUTED BY DIFFUN. IT IS ALSO ACCESSED AS                   * GEAR 530
      C*        SAVE(N2,1) AS A COMPLETE ARRAY.                              * GEAR 540
      C*    SAVE(N5+I,1) HOLDS THE DERIVATIVES DURING JACOBIAN EVALUATIONS. * GEAR 550
      C*        IT IS REFERENCED AS SAVE(N6,1) AS A COMPLETE ARRAY.          * GEAR 560
      C*    THE PARAMETERS TO THE SUBROUTINE DIFSUB  HAVE                    * GEAR 570
      C*    THE FOLLOWING MEANIGS..                                          * GEAR 580
      C*     N     THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS. N       * GEAR 590
      C*        MAY BE DECREASED ON LATER CALLS IF THE NUMBER OF             * GEAR 600
            CONTINUE                                                          GEAR 610
      C*        ACTIVE EQUATIONS REDUCES, BUT IT MUST NOT BE                 * GEAR 620
      C*        INCREASED WITHOUT CALLING WITH JSTART = 0.                   * GEAR 630
      C*       THE INDEPENDENT VARIABLE.                                     * GEAR 640
      C*       AN 8 BY N ARRAY CONTAINING THE DEPENDENT VARIABLES AND        * GEAR 650
      C*        THEIR SCALED DERIVATIVES.  Y(J+1,I) CONTAINS                 * GEAR 660
      C*        THE J-TH DERIVATIVE OF Y(I) SACALED BY                       * GEAR 670
      C*        H**J/FACTORIALNJE WHERE H IS THE CURRENT                     * GEAR 680
      C*        STEP SIZE. ONLY YN1,IE NEED BE PROVIDED BY                   * GEAR 690
      C*        THE CALLING PROGRAM ON THE FIRST ENTRY.                      * GEAR 700
      C*         IF IT IS DESIRED TO INTERPOLATE TO NON MESH POINTS          * GEAR 710
```

```
C*         THESE VALUES CAN BE USED. IF THE CURRENT STEP SIZE             * GEAR 720
C*         IS H AND THE VALUE AT T + E IS NEEDED, FORM                    * GEAR 730
C*         S = E/H, AND THEN COMPUTE                                      * GEAR 740
C*               NQ                                                       * GEAR 750
C*          Y(I) AT+E) = SUM Y(J+1,I)*S**J                                * GEAR 760
C*                             J=0                                        * GEAR 770
C*     SAVE  A BLOCK OF AT LEAST 12*N FLOATING POINT LOCATIONS            * GEAR 780
C*           USED BY THE SUBROUTINES.                                     * GEAR 790
C*         THE STEP SIZE TO BE BE ATTEMPTED ON THE NEXT STEP.             * GEAR 800
C*          H MAY BE ADJUSTED UP OR DOWN BY THE PROGRAM                   * GEAR 810
C*          IN ORDER TO ACHEIVE AN ECONOMICAL INTEGRATION.                * GEAR 820
C*          HOWEVER, IF THE H PROVIDED BY THE USER DOES                   * GEAR 830
C*          NOT CAUSE A LARGER ERROR THA REQUESTED, IT                    * GEAR 840
C*          WILL BE USED. TO SAVE COMPUTER TIME, THE USER IS              * GEAR 850
C*          ADVISED TO USE A FAIRLY SMALL STEP FOR THE FIRST              * GEAR 860
C*          CALL. IT WILL BE AUTOMATICALLY INCREASED LATER.               * GEAR 870
C*     HMIN  THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE              * GEAR 880
       CONTINUE                                                             GEAR 890
C*          INTEGRATION. NOTE THAT ON STARTING THIS MUST                  * GEAR 900
C*          MUCH SMALLER THAN THE AVERAGE H EXPECTED SINCE                * GEAR 910
C*          A FIRST ORDER METHOD IS USED INITIALLY.                       * GEAR 920
C*     HMAX   THE MAXIMUM SIZE TO WHICH THE STEP WILL BE INCREASED        * GEAR 930
C*     EPS    THE ERROR TEST CONSTANT. SINGLE STEP ERROR ESTIMATES        * GEAR 940
C*             DIVIDED BY WMAX(I) MUST BE LESS THAN THIS                  * GEAR 950
C*             ADJUSTED TO ACHEIVE THIS.                                  * GEAR 960
C*     MF     THE METHOD INDICATOR. THE FOLLOWING ARE ALLOWED..           * GEAR 970
C*             0  AN ADAMAS PREDICTOR CORRECTOR IS USED.                  * GEAR 980
C*             1  A MULTI-STEP METHOD SUITABLE FOR STIFF                  * GEAR 990
C*                  SYSTEMS IS USED. IT WILL ALSO WORK FOR                * GEAR1000
C*                  NON STIFF SYSTEMS. HOWEVER THE USER                   * GEAR1010
C*                  MUST PROVIDE A SUBROUTINE PEDERV WHICH                * GEAR1020
C*                  EVALUATES THE PARTIAL DERIVATIVES OF                  * GEAR1030
C*                  THE DIFFERENTIAL EQUATIONS WITH RESPECT               * GEAR1040
C*                  TO THE YHS.  THIS IS DONE BY CALL                     * GEAR1050
C*                  PEDERVNT,Y,PW,ME PW IS AN N BY N ARRAY                * GEAR1060
C*                  WHICH MUST BE SET TO THE PARTIAL OF                   * GEAR1070
```

```
C*                 THE I-TH EQUATION WITH RESPECT                    * GEAR1080
C*                 TO THE J DEPENDENT VARIABLE IN PW(I,J).           * GEAR1090
        CONTINUE                                                       GEAR1100
C*                 PW IS ACTUALLY STORED IN AN M BY M                * GEAR1110
C*                 ARRAY WHERE M IS THE VALUE OF N USED ON           * GEAR1120
C*                 THE FIRST CALL TO THIS PROGRAM.                   * GEAR1130
C*                 2 THE SAME AS CASE 1, EXCEPT THAT THIS            * GEAR1140
C*                   SUBROUTINE COMPUTES THE PARTIAL                 * GEAR1150
C*                 DERIVATIVES BY NUMERICAL DIFFERENCING             * GEAR1160
C*                   OF THE DERIVATIVES. HENCE PEDERV IS             * GEAR1170
C*                   NOT CALLED.                                     * GEAR1180
C*      YMAX  AN ARRAY OF N LOCATIONS WHICH CONTAINS THE MAXIMUM     * GEAR1190
C*          OF EACH Y SEEN SO FAR.  IT SHOULD NORMALL Y BE SET TO    * GEAR1200
C*          1 IN EACH COMPONENT BEFORE THE FIRST ENTRY. (SEE THE     * GEAR1210
C*          DESCRIPTION OF EPS.)                                     * GEAR1220
C*      ERROR AN ARRAY OF N ELEMENTS WHICH CONTAINS THE ESTIMATED    * GEAR1230
C*        ONE STEP ERROR IN EACH COMPONENT.                          * GEAR1240
C*      KFLAG  A COMPLETION CODE WITH THE FOLLOWING MEANINGS..       * GEAR1250
C*             +1 THE STEP WAS SUCCESFUL.                            * GEAR1260
        CONTINUE                                                       GEAR1270
C*             -1 THE STEP WAS TAKEN WITH H = HMIN, BUT THE          * GEAR1280
C*                REQUESTED ERROR WAS NOT ACHIEVED.                  * GEAR1290
C*             -2 THE MAXIMUM ORDER SPECIFIED WAS FOUND TO           * GEAR1300
C*                BE TOO LARGE.                                      * GEAR1310
C*             -3 CORRECTOR CONVERGENCE COULD NOT BE                 * GEAR1320
C*                ACHIEVED FOR H .GT. HMIN.                          * GEAR1330
C*             -4 THE REQUESTED ERROR IS SMALLER THAN CAN            * GEAR1340
C*                BE HANDLED FOR THIS PROBLEM.                       * GEAR1350
C*      JSTART AN INPUT INDICATOR WITH THE FOLLOWING MEANINGS..      * GEAR1360
C*             -1  REPEAT THE LAST STEP WITH A NEW H                 * GEAR1370
C*             0 PERFORM THE FIRST STEP. THE FIRST STEP              * GEAR1380
C*                MUST BE DONE WITH THIS VALUE OF JSTART             * GEAR1390
C*                SO THAT THE SUBROUTINE CAN INITIALIZE              * GEAR1400
C*                ITSELF.                                            * GEAR1410
C*             +1 TAKE A NEW STEP CONTINUING FROM THE LAST.          * GEAR1420
        CONTINUE                                                       GEAR1430
```

```
C*        JSTART IS SET TO NQ. THE CURRENT ORDER OF THE METHOD       * GEAR1440
C*        AT EXIT. NQ IS ALSO THE ORDER OF THE MAXIMUM               * GEAR1450
C*        DERIVATIVE AVAILABLE.                                      * GEAR1460
C*     MAXDER THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE      * GEAR1470
C*        METHOD. SINCE THE ORDER IS EQUAL TO THE HIGHEST            * GEAR1480
C*        DERIVATIVE USED, THIS RESTRICTS THE ORDER. IT MUST         * GEAR1490
C*        BE LESS THAN 8 FOR ADMAS AND 7 FOR STIFF METHODS.          * GEAR1500
C*       A BLOCK OF AT LEAST N**2 FLOATING POINT LOCATIONS.          * GEAR1510
C*     ************************************************************** GEAR1520
        DIMENSION Y(8,1),YMAX(1),SAVE(10,1),ERROR(1),PW(1),             GEAR1530
     1     A(8),PERTST (7,2,3)                                          GEAR1540
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ GEAR1550
       DIMENSION PPW(1)                                                 GEAR1560
       DIMENSION LLL(1),MMM(1)                                          GEAR1570
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ GEAR1580
C*******************************************************************  GEAR1590
C*     THE COEFFICIENTS IN PERTST ARE USED IN SELECTING THE STEP AND * GEAR1600
C*     ORDER, THEREFORE ONLY ABOUT ONE PERCENT ACCURACY IS NEEDED.   * GEAR1610
C*********************************************************************GEAR1620
       DATA PERTST /2.0,4.5,7.333,10.42,13.7,17.15,1.0,                 GEAR1630
     1             2.0,12.0,24.0,37.89,53.33,70.08,87.97,               GEAR1640
     2             3.0,6.0,9.167,12.5,15.98,1.0,1.0,                    GEAR1650
     3             12.0,24.0,37.89,53.33,70.08,87.97,1.0,               GEAR1660
     4             1.,1.,0.5,0.1667,0.04133,0.008267,1.0,               GEAR1670
     5             1.0,1.0,2.0,1.0,.3157, .07407,.0139/                 GEAR1680
       DATA A(2) / -1.0/                                                GEAR1690
       IRET = 1                                                         GEAR1700
       KFLAG = 1                                                        GEAR1710
       IF (JSTART.LE.0)GO TO 140                                        GEAR1720
C*********************************************************************GEAR1730
C* BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING      GEAR1740
C* H H BY THE FARTOR R IF THE CALLER HAS CHANGED H. ALLL VARIABLES     GEAR1750
C* DEPENDENT ON H MUST ALSO BE CHANGED                                 GEAR1760
C* E IS A COMPARISON FOR ARRORS OF THE CURRENT ORDER NQ. EUP IS        GEAR1770
C* TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE ORDER.    GEAR1780
C* HNEW 2S THE STEP SIZE THAT WAS USED ON THE LAST CALL.               GEAR1790
```

```
C**********************************************************************GEAR1800
  100 DO 110 I = 1,N                                                    GEAR1810
      DO 110 J= 1,K                                                     GEAR1820
  110 SAVE(J,I) = Y(J,I)                                                GEAR1830
      HOLD = HNEW                                                       GEAR1840
      IF(H.EQ.HOLD) GO TO 130                                           GEAR1850
  120 RACUM = H/HOLD                                                    GEAR1860
      IRET1 = 1                                                         GEAR1870
      GO TO 750                                                         GEAR1880
  130 NQOLD = NQ                                                        GEAR1890
      TOLD = T                                                          GEAR1900
      RACUM = 1.0                                                       GEAR1910
      IF(JSTART.GT.0) GO TO 250                                         GEAR1920
      GO TO 170                                                         GEAR1930
  140 IF(JSTART.EQ. -1) GO TO 160                                       GEAR1940
C**********************************************************************GEAR1950
C* ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL             GEAR1960
C* DERIVATIVES ARE CALCULATED.                                          GEAR1970
C**********************************************************************GEAR1980
      NQ = 1                                                            GEAR1990
      N3 = N                                                            GEAR2000
       N1 = N*10                                                        GEAR2010
      N2 = N1 + 1                                                       GEAR2020
      N4 = N**2                                                         GEAR2030
      N5 = N1 + N                                                       GEAR2040
      N6 = N5 + 1                                                       GEAR2050
      CALL DIFFUN(T,Y,SAVE(N2,1))                                       GEAR2060
      DO 150 I = 1,N                                                    GEAR2070
  150 Y(2,I) = SAVE(N1 + I,1)*H                                         GEAR2080
      HNEW = H                                                          GEAR2090
      K = 2                                                             GEAR2100
      GO TO 100                                                         GEAR2110
C**********************************************************************GEAR2120
C* REPEAT LAST STEP BY RESTORING SAVED INFORMATION.                     GEAR2130
C**********************************************************************GEAR2140
  160 IF(NQ.EQ.NQOLD) JSTART = 1                                        GEAR2150
```

```
      T = TOLD                                                          GEAR2160
      NQ = NQOLD                                                        GEAR2170
      K = NQ + 1                                                        GEAR2180
      GOTO 120                                                          GEAR2190
C***********************************************************************GEAR2200
C* SET THE COEFFICIENTS THAT DETERMINE THE ORDER AND THE METHOD          GEAR2210
C* TYPE, CHECK FOR EXEESSIVE ORDER. THE LAST TWO STATEMENTS OF          GEAR2220
C* THIS SECTION SET IWEVAL .GT. 0 IF PW IS TO BE REEVALUATED            GEAR2230
C* BECAUSE OF THE ORDER CHANGE, AND THEN REPEAT THE INTEGRATION         GEAR2240
C* STEP IF IT HAS NOT BEEN DONE (IRET = 1) OR SKIP TO A FINAL           GEAR2250
C* SCALING BEFORE EXIT IF IT HAS BEEN COMPLETED (IRET = 2).             GEAR2260
C***********************************************************************GEAR2270
 170  IF(MF.EQ.0) GO TO 180                                             GEAR2280
      IF(NQ.GT.6) GO TO 190                                             GEAR2290
      GO TO(221,222,223,224,225,226),NQ                                 GEAR2300
 180  IF( NQ.GT.7) GO TO 190                                            GEAR2310
      GO TO (211,212,213,214,215,216,217),NQ                            GEAR2320
 190  KFLAG = -2                                                        GEAR2330
      RETURN                                                            GEAR2340
C***********************************************************************GEAR2350
C* THE FOLLOWING COEFFICIENTS SHOULD BE BEFINED TO THE MAXIMUM          GEAR2360
C* ACCURACY PERMITTED BY THE MACHINE. THEY ARE, IN IN THE ORDER USED..  GEAR2370
C*                                                                      GEAR2380
C* -1                                                                   GEAR2390
C* -1/2,-1/2                                                            GEAR2400
C* -5/12,-3/4,-1/6                                                      GEAR2410
C* -3/8,-11/12,-1/3,-1/24                                               GEAR2420
C* -251/720,-25/24,-35/72,-5/48,-1/120                                  GEAR2430
C* -95/288,-137/120,-5/8-17/96,-1/40,-1/720                             GEAR2440
C* -19087/60480,-49/40,-203/270,-49/192,-7/144,-7/1440,-1/5040          GEAR2450
C*                                                                      GEAR2460
C* 1                                                                    GEAR2470
C* -2/3,-1/                                                             GEAR2480
C* -6/11,-6/11,-1/11                                                    GEAR2490
C* -12/25,-7/10,-1/5,-1/50                                              GEAR2500
C* -120/274,-225/274,-85/274,-15/274,-1/274                             GEAR2510
```

```
C* -180/441,-58/63,-15/36,-25/252,-3/252,-1/1764                        GEAR2520
C***********************************************************************GEAR2530
 211   A(1) = -1.0                                                      GEAR2540
       GO TO 230                                                        GEAR2550
 212   A(1) = -0.50000000                                               GEAR2560
       A(3)   = -0.50000000                                             GEAR2570
       GO TO 230                                                        GEAR2580
 213   A(1) =  -0.41666666666667                                        GEAR2590
       A(3) = -0.75000000                                               GEAR2600
       A(4) = -0.16666666666666667                                      GEAR2610
       GO TO 230                                                        GEAR2620
 214   A(1) = -0.37500000                                               GEAR2630
       A(3) = -0.91666666666666667                                      GEAR2640
       A(4) = -0.33333333333333333                                      GEAR2650
       A(5) = - 0.041666666666666667                                    GEAR2660
       GO TO 230                                                        GEAR2670
 215   A(1) = -0.34861111111111111                                      GEAR2680
       A(3) = -1.0416666666666667                                       GEAR2690
       A(4) = -0.48611111111111111                                      GEAR2700
       A(5) = - 0.10416666666666667                                     GEAR2710
       A(6) = -0.0083333333333333333                                    GEAR2720
       GO TO 230                                                        GEAR2730
 216   A(1) = -0.32986111111111111                                      GEAR2740
       A(3) = -1.1416666666666667                                       GEAR2750
       A(4) = -0.62500000                                               GEAR2760
       A(5) = - 0.17708333333333333                                     GEAR2770
       A(6) = -0.025000000                                              GEAR2780
       A(7) = -0.0013888888888888889                                    GEAR2790
       GO TO 230                                                        GEAR2800
 217   A(1) = -0.3155919312169312                                       GEAR2810
       A(3) = -1.23500000                                               GEAR2820
       A(4) = -0.7518518518518519                                       GEAR2830
       A(5) = -0.2552083333333333                                       GEAR2840
       A(6) = - 0.04861111111111111                                     GEAR2850
       A(7) = - 0.004861111111111111                                    GEAR2860
       A(8) = -0.0001984126984126984                                    GEAR2870
```

```
      GO TO 230                                                         GEAR2880
  221 A(1) = -1.00000000                                                GEAR2890
      GO TO 230                                                         GEAR2900
  222 A(1) = -0.6666666666666667                                        GEAR2910
      A(3) = -0.3333333333333333                                        GEAR2920
      GO TO 230                                                         GEAR2930
  223 A(1) = -0.5454545454545455                                        GEAR2940
      A(3) = A(1)                                                       GEAR2950
      A(4) = -.09090909090909091                                        GEAR2960
      GO TO 230                                                         GEAR2970
  224 A(1) = -0.48000000                                                GEAR2980
      A(3) = -0.70000000                                                GEAR2990
      A(4) = -0.20000000                                                GEAR3000
      A(5) = -0.02000000                                                GEAR3010
      GO TO 230                                                         GEAR3020
  225 A(1) =-0.437956204379562                                          GEAR3030
      A(3) = -0.8211678832116788                                        GEAR3040
      A(4) = -0.3102189781021898                                        GEAR3050
      A(5) = -0.05474452554744526                                       GEAR3060
      A(6) = -0.0036496350364963504                                     GEAR3070
      GO TO 230                                                         GEAR3080
  226 A(1) = -0.4081632653061225                                        GEAR3090
      A(3) = -0.9206349206349206                                        GEAR3100
      A(4) = -0.4166666666666667                                        GEAR3110
      A(5) = -0.0992063492063492                                        GEAR3120
      A(6) = -0.0119047619047619                                        GEAR3130
      A(7) = -0.0005668934240362 82                                     GEAR3140
  230 K = NQ + 1                                                        GEAR3150
      IDOUB = K                                                         GEAR3160
      MTYP = (4 - MF)/2                                                 GEAR3170
      ENQ2 = .5/FLOAT(NQ + 1)                                           GEAR3180
      ENQ3 = .5/FLOAT(NQ + 2)                                           GEAR3190
      ENQ1 = 0.5/FLOAT(NQ)                                              GEAR3200
      PEPSH = EPS                                                       GEAR3210
      EUP = (PERTST(NQ,MTYP,2)*PEPSH)**2                                GEAR3220
      E = (PERTST(NQ,MTYP,1)*PEPSH)**2                                  GEAR3230
```

```
      EDWN = (PERTST(NQ,MTYP,3)*PEPSH)**2                               GEAR3240
      IF (EDWN.EQ.0) GO TO 780                                          GEAR3250
      BND = EPS*ENQ3/DFLOAT(N)                                          GEAR3260
  240 IWEVAL = MF                                                       GEAR3270
      GO TO (250,680),IRET                                              GEAR3280
C***********************************************************************GEAR3290
C* THES SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY            GEAR3300
C* MULTIPLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE             GEAR3310
C* MATRIX                                                               GEAR3320
C***********************************************************************GEAR3330
  250 T = T + H                                                         GEAR3340
      DO 260 J= 2,K                                                     GEAR3350
      DO 260 J1 = J,K                                                   GEAR3360
      J2 = K - J1 + J-1                                                 GEAR3370
      DO 260 I= 1,N                                                     GEAR3380
  260 Y(J2,I) = Y(J2,I) + Y(J2+1,I)                                     GEAR3390
C***********************************************************************GEAR3400
C* UP TO 3 CORRECTOR ITERATIONS ARE TAKEN. CONVERGENCE IS TESTED        GEAR3410
C* BY REQUIRING CHANGES TO BE LESS THAN BND WHICH IS DEPENDENT ON       GEAR3420
C* THE ERROR TEST CONSTANT                                              GEAR3430
C*   THE SUM OF THE CORRECTIONS IS ACCUMULATED IN THE ARRAY             GEAR3440
C* ERROR(I). IT IS EQUAL TO THE K-YH DERIVATIVE OF Y MULTIPLIED         GEAR3450
C* BY H**K/(FACTORIAL(K-1)*A(K)),AND IS THEREFORE PROPORTIONAL          GEAR3460
C* TO THE ACTUAL ERRORS TO THE LOWEST POWER OF H PRESENT. 'H**K'        GEAR3470
C***********************************************************************GEAR3480
      DO 270 I=1,N                                                      GEAR3490
  270 ERROR(I) = 0.0                                                    GEAR3500
      DO 430 L=1,3                                                      GEAR3510
      CALL DIFFUN(T,Y,SAVE(N2,1))                                       GEAR3520
       NK = N**2                                                        GEAR3530
C***********************************************************************GEAR3540
C* IF THERE HAS BEEB A CHANGE OF ORDER OR OR THERE HAS BEEN TROUBLE     GEAR3550
C* WITH C3NVERGENCE,PW IS RE-EVALUATED PRIOR TO STARTING THE            GEAR3560
C* CORRECTOR ITERATION IN THE CASE OF STIFF METHODS. IWEVAL IS          GEAR3570
C* THEN SET TO -1 AS AN INDICATOR THAT IT HAS BEEN DONE.                GEAR3580
C***********************************************************************GEAR3590
```

```
      IF(IWEVAL.LT.1) GO TO 350                                         GEAR3600
      IF(MF.EQ.2) GO TO 310                                             GEAR3610
      CALL PEDERV(T,Y,PW,N3)                                            GEAR3620
      R = A(1)*H                                                        GEAR3630
      DO 280 I=1,N4                                                     GEAR3640
280   PW(I) = PW(I)*R                                                   GEAR3650
C***********************************************************************GEAR3660
C* ADD THE IDENTITY MATRIX TO THE JACOBIAN AND INVERT TO GET PW.        GEAR3670
C***********************************************************************GEAR3680
290   DO 300 I=1,N                                                      GEAR3690
300   PW(I*(N3+1)-N3) = 1.0 + PW(I*(N3+1)-N3)                           GEAR3700
      IWEVAL = -1                                                       GEAR3710
C ---------------------------------------------------------------------- GEAR3720
C THE CARDS INCLUDED BETWEEN THE TWO ALL '$' CARDS WERE ADDED IN        GEAR3730
C ORDER TO MATCH THE CALLING OF THE MATRIX INVERSION SUBROUTINE         GEAR3740
C TO THE MINV SUBROUTINE OF THE  IBM SCIENTIFIC SUBROUTINE PACKAGE      GEAR3750
C ORIGINALLY THE GEARSB PROGRAM HAD THE FOLLOWING CARD                  GEAR3760
C     CALL MATINV(PW,N,N3,J1)                                           GEAR3770
C ---------------------------------------------------------------------- GEAR3780
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ GEAR3790
      DO 1000 I = 1,NK                                                  GEAR3800
 1000 PPW(I) = PW(I)                                                    GEAR3810
      CALL MINV(PPW,N,PDET,LLL,MMM)                                     GEAR3820
      IF( ABS(PDET).LE. 1.E-05)PDET = 0.E00                             GEAR3830
      IF(PDET.EQ.0.E00) JI = -1                                         GEAR3840
C ######################################################################GEAR3850
      IF(PDET .EQ. 0.E00) WRITE(6,1600)                                 GEAR3860
 1600 FORMAT(1X,'SINGULAR MATRIX')                                      GEAR3870
C ######################################################################GEAR3880
      IF(PDET.NE.0.E00) JI = +1                                         GEAR3890
       DO 1100 I = 1,NK                                                 GEAR3900
 1100 PW(I) = PPW(I)                                                    GEAR3910
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ GEAR3920
      IF(J1.GT.0)GO TO 350                                              GEAR3930
      GO TO 440                                                         GEAR3940
C***********************************************************************GEAR3950
```

```
C*********************************************************************GEAR3960
C* EVALUATE THE JACOBIAN INTO PW BY NUMERICAL DIFFERENCING. R IS THE   GEAR3970
C* CHANGE MADE TO THE ELEMENT OF Y.IT IS EPS RELATIVE TO Y WITH         GEAR3980
C* A MINIMUM OF EPS**2                                                  GEAR3990
C*********************************************************************GEAR4000
  310   DO 320 I=1,N                                                     GEAR4010
  320   SAVE(9,I) = Y(1,I)                                               GEAR4020
        DO 340 J=1,N                                                     GEAR4030
        R = EPS*DMAX1(EPS,DABS(SAVE(9,J)))                               GEAR4040
        Y(1,J) = Y(1,J) + R                                              GEAR4050
        D = A(1)*H/R                                                     GEAR4060
        CALL DIFFUN(T,Y,SAVE(N6,1))                                      GEAR4070
        DO 330 I = 1,N                                                   GEAR4080
  330   PW(I+(J-1)*N3) =  (SAVE(N5+I,1) - SAVE(N1+I,1))*D                GEAR4090
  340   Y(1,J) = SAVE(9,J)                                               GEAR4100
        GO TO 290                                                        GEAR4110
  350   IF(MF.NE.0) GO TO 370                                            GEAR4120
        DO 360 I=1,N                                                     GEAR4130
  360   SAVE(9,I)= Y(2,I) - SAVE(N1+I,1)*H                               GEAR4140
        GO TO 410                                                        GEAR4150
  370   DO 380 I= 1,N                                                    GEAR4160
  380   SAVE(N5+I,1) =  Y(2,I) - SAVE(N1+I,1)*H                          GEAR4170
        DO 400 I=1,N                                                     GEAR4180
        D = 0.0                                                          GEAR4190
        DO 390 J=1,N                                                     GEAR4200
  390   D = D + PW(I+(J-1)*N3)*SAVE(N5+J,1)                              GEAR4210
   400  SAVE(9,I) = D                                                   GEAR4220
  410   NT = N                                                           GEAR4230
C*********************************************************************GEAR4240
C* CORRECT AND SEE IF ALL CHANGES ARE LESS THAN BND RELATIVE TO YMAX.   GEAR4250
C* IF SO,  HE CORRECTOR IS SAID TO HAVE CONVERGED.                      GEAR4260
C*********************************************************************GEAR4270
        DO 420 I=1,N                                                     GEAR4280
        Y(1,I) = Y(1,I) + A(1)*SAVE(9,I)                                 GEAR4290
        Y(2,I) = Y(2,I) - SAVE(9,I)                                      GEAR4300
        ERROR(I) = ERROR(I) + SAVE(9,I)                                  GEAR4310
```

```
      IF(DABS(SAVE(9,I)).LE.(BND*YMAX(I))) NT = NT - 1                  GEAR4320
  420 CONTINUE                                                          GEAR4330
      IF (NT.LE.0) GO TO 490                                            GEAR4340
  430 CONTINUE                                                          GEAR4350
C***********************************************************************GEAR4360
C* THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES. VARIOUS       GEAR4370
C* POSSIBILITIES ARE CHECKED FOR. IF H IS ALREADY HMIN AND              GEAR4380
C* THIS IS EITHER ADAMS METHOD OR THE STIFF METHOD IN WHICH THE         GEAR4390
C* MATRIX PW HAS ALREADY BEEN EVALUATED, A NO CONVERGENCE EXIT          GEAR4400
C* IS TAKEN. OTHERWISE THE MATRIX PW IS RE-EVALUATED AND/OR THE         GEAR4410
C* STEP IS REDUCED TO TRY AND GET CONVERGENCE.                          GEAR4420
C***********************************************************************GEAR4430
  440 T = TOLD                                                          GEAR4440
      IF((H.LE.(HMIN*1.00001)).AND.((IWEVAL - MTYP).LT.-1)) GO TO 460   GEAR4450
      IF ((MF.EQ.0).OR.(IWEVAL.NE.0))RACUM = RACUM*0.25D0               GEAR4460
      IWEVAL = MF                                                       GEAR4470
      IRET1 = 2                                                         GEAR4480
      GO TO 750                                                         GEAR4490
  460 KFLAG = -3                                                        GEAR4500
  470 DO 480 I=1,N                                                      GEAR4510
      DO 480 J=1,K                                                      GEAR4520
  480 Y(J,I) = SAVE(J,I)                                                GEAR4530
      H = HOLD                                                          GEAR4540
      NQ = NQOLD                                                        GEAR4550
      JSTART = NQ                                                       GEAR4560
      RETURN                                                            GEAR4570
C***********************************************************************GEAR4580
C* THE CORRECTOR CONVERGED AND CONTROL IS PASSED TO STATEMENT 520       GEAR4590
C* IF THE ERROR TEST IS O.K., AND TO 540 OTHERWISE.                     GEAR4600
C* IF THE STEP IS O.K. IT IS ACCEPTED. IF IDOUB HAS BEEN REDUCED        GEAR4610
C* TO ONE, A TEST IS MADE TO SEE TO SEE IF THE STEP CAN BE INCREASED    GEAR4620
C* AT THE CURRENT ORDER BY GOING TO ONE HIGHER OR ONE LOWER.            GEAR4630
C* SUCH A CHANGE IS ONLY MADE IF THE STEP CAN BE INCREASED BY AT        GEAR4640
C* LEAST 1.1. IF NO CHANGE IS POSSIBLE IDOUB IS SET TO 10 TO            GEAR4650
C* PREVENT FURTHER TESTING FOR 10 STEPS.                                GEAR4660
C* IF A CHANGE IS POSSIBLE, IT IS MADE AND IDOUB IS SET TO              GEAR4670
```

```
C* NQ + 1 TO PREVENT FURTHER TESTING FOR THAT NUMBER OF STEPS.          GEAR4680
C* IF THE ERROR WAS TOO LARGE, THE OPTIMUM STEP SIZE FOR THIS OR        GEAR4690
C* LOWER ORDER IS COMPUTED, AND THE STEP RETRIED. IF IT SHOULD          GEAR4700
C* FAIL TWICE MOREIT IS AN INDICATION THAT THE DERIVATIVES THAT HAVE ACCGEAR4710
C* UMULATED IN THE Y ARRAY HAVE HAVE ERRORS OF THE WRONG ORDER          GEAR4720
C* SO THE FIRST DERIVATIVES ARE RECOMPUTED AND THE ORDER IS SET TO 1.   GEAR4730
C***********************************************************************GEAR4740
  490 D = 0.0                                                           GEAR4750
      DO 500 I=1,N                                                      GEAR4760
  500 D = D + (ERROR(I)/YMAX(I))**2                                     GEAR4770
      IWEVAL = 0                                                        GEAR4780
      IF(D.GT.E) GO TO 540                                              GEAR4790
      IF(K.LT.3)GO TO 520                                               GEAR4800
C***********************************************************************GEAR4810
C* COMPLETE THE CORRECTION OF THE HIGHER ORDER DERIVATIVES AFTER A      GEAR4820
C* SUCCESFUL STEP                                                       GEAR4830
C***********************************************************************GEAR4840
      DO 510 J=3,K                                                      GEAR4850
      DO 510 I=1,N                                                      GEAR4860
  510 Y(J,I) = Y(J,I) + A(J)*ERROR(I)                                   GEAR4870
  520 KFLAG = +1                                                        GEAR4880
      HNEW = H                                                          GEAR4890
      IF(IDOUB.LE.1) GO TO 550                                          GEAR4900
      IDOUB = IDOUB -1                                                  GEAR4910
      IF(IDOUB.GT.1) GO TO 700                                          GEAR4920
      DO 530 I=1,N                                                      GEAR4930
  530 SAVE(10,I) = ERROR(I)                                             GEAR4940
      GO TO 700                                                         GEAR4950
C***********************************************************************GEAR4960
C* REDUCE THE FAILURE FLAG COUNT TO CHECK FOR MULTIPLE FAILURES.        GEAR4970
C* RESTORE   TO ITS ORIGINAL VALUE AND TRY AGAIN UNLESS THERE HAVE      GEAR4980
C* THREE FAILURES. IN THAT CASE THE DERIVATIVES ARE ASSUMED TO HAVE     GEAR4990
C* ACCUMULATED ERRORS SO A RESTART FROM THE CURRENT VALUES OF Y IS      GEAR5000
C* TRIED                                                                GEAR5010
C***********************************************************************GEAR5020
  540 KFLAG = KFLAG - 2                                                 GEAR5030
```

```
      IF(H.LE.(HMIN*1.CCCC1)) GC TO 74C                                GEAR5040
      T = TCLC                                                          GEAR5C5C
      IF(KFLAG .LE. -5) GC TC 720                                       GEAR5C6C
C***********************************************************************GEAR5C7C
C* PR1,PR2,PR3 WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE          GEAR5080
C* SHCLLD  E CIVICEC AT CRCER CNE LCWER, AT THES CRCER, ANC AT ORDER    GEAR5C9C
C* CNE HIGHER RESPECTIVELY                                              GEAR51CC
C***********************************************************************GEAR5110
 550  PR2 = (C/E)**ENC2*1.2                                             GEAR512C
      PR3 = 1.E+2C                                                      GEAR513C
      IF((NG.GE. MAXDER).OR.(KFLAG.LE. -1)) GC TO 570                   GEAR5140
      C = C.0                                                           GEAR515C
      CC 56C I=1,N                                                      GEAR516C
 560  D=D + ((ERRCR(I) - SAVE(1C,I))/YMAX(I))**2                        GEAR517C
      PR3 = (D/EUP)**ENC3*1.4                                           GEAR518C
 570  PR1 = 1.E+20                                                      GEAR519C
      IF(NQ.LE.1) GO TC 59C                                             GEAR52CC
      C = 0.0                                                           GEAR5210
      CC 58C I = 1,N                                                    GEAR522C
 580  C = D + (Y(K,I)/YMAX(I))**2                                       GEAR5230
      PR1 = (D/ECWN)**ENQ1*1.3                                          GEAR5240
 590  CCNTINUE                                                          GEAR525C
      IF(PR2.LE.PR3) GC TC 650                                          GEAR526C
      IF (PR3.LT.PR1) GO TO 66C                                         GEAR5270
 600  R = 1.0/AMAX1(PR1,1.E-4)                                          GEAR528C
      NEWC = NC - 1                                                     GEAR529C
 610   IDCLB = 1C                                                       GEAR53CC
      IF((KFLAC.EQ.1).AND.(R.LT.(1.1))) GO TO 7CC                       GEAR5310
      IF(NEWC.LE.NC) GC TC 630                                          GEAR532C
C***********************************************************************GEAR533C
C* CCMPUTE CNE ACCITIONAL SCALED DERIVATIVE IF CRCER IS INCREASED       GEAR5340
C***********************************************************************GEAR535C
      CC 62C I = 1,N                                                    GEAR536C
 620  Y(NEWC+1,I) = ERROR(I)*A(K)/DFLOAT(K)                             GEAR5370
 630  K = NEWC+1                                                        GEAR538C
      IF(KFLAG.EC.1) GC TC 670                                          GEAR539C
```

```
      RACUM = RACUM*R                                                   GEAR5400
      IRET1 = 3                                                         GEAR5410
      GO TO 750                                                         GEAR5420
  640 IF(NEWQ.EQ.NQ) GO TO 250                                          GEAR5430
      NQ = NEWQ                                                         GEAR5440
      GO TO 170                                                         GEAR5450
  650 IF(PR2.GT.PR1) GO TO 600                                          GEAR5460
      NEWQ = NQ                                                         GEAR5470
      R = 1.0/AMAX1(PR2,1.E-4)                                          GEAR5480
      GO TO 610                                                         GEAR5490
  660 R = 1.0/AMAX1(PR3,1.E-4)                                          GEAR5500
      NEWQ = NQ + 1                                                     GEAR5510
      GO TO 610                                                         GEAR5520
  670 IRET = 2                                                          GEAR5530
      R = DMIN1(R,HMAX/DABS(H))                                         GEAR5540
      H = H*R                                                           GEAR5550
      HNEW = H                                                          GEAR5560
      IF(NQ.EQ.NEWQ) GO TO 680                                          GEAR5570
      NQ = NEWQ                                                         GEAR5580
      GO TO 170                                                         GEAR5590
  680 R1 = 1.0                                                          GEAR5600
      DO 690 J = 2,K                                                    GEAR5610
      R1 = R1*R                                                         GEAR5620
      DO 690 I = 1,N                                                    GEAR5630
  690 Y(J,I) = Y(J,I)*R1                                                GEAR5640
      IDOUB = K                                                         GEAR5650
  700 DO 710 I=1,N                                                      GEAR5660
  710 YMAX(I) = DMAX1(YMAX(I),DABS(Y(1,I)))                             GEAR5670
      JSTART = NQ                                                       GEAR5680
      RETURN                                                            GEAR5690
  720 IF(NQ.EQ.1) GO TO 780                                             GEAR5700
      CALL DIFFUN(T,Y,SAVE(N2,1))                                       GEAR5710
      R = H/HOLD                                                        GEAR5720
      DO 730 I = 1,N                                                    GEAR5730
      Y(1,I) = SAVE(1,I)                                                GEAR5740
      SAVE(2,I) = HOLD*SAVE(N1+I,1)                                     GEAR57
```

```
      730   Y(2,I) = SAVE(2,I)*R                                                GEAR5760
            NQ = 1                                                              GEAR5770
            KFLAG = 1                                                           GEAR5780
            GO TO 170                                                           GEAR5790
      740   KFLAG = -1                                                          GEAR5800
            HNEW = H                                                            GEAR5810
            JSTART = NQ                                                         GEAR5820
            RETURN                                                              GEAR5830
      C*************************************************************************GEAR5840
      C* THIS SECTION SCALES ALL VARIABLES CONNECTED WITH H AND RETURNS         GEAR5850
      C* TO THE ENTERING SECTION                                                GEAR5860
      C*************************************************************************GEAR5870
      750   RACUM = DMAX1(DABS(HMIN/HOLD),RACUM)                                GEAR5880
            RACUM = DMIN1(RACUM,DABS(HMAX/HOLD))                                GEAR5890
            R1 = 1.0                                                            GEAR5900
            DO 760 J = 2,K                                                      GEAR5910
            R1 = R1*RACUM                                                       GEAR5920
            DO 760 I = 1,N                                                      GEAR5930
      760   Y(J,I) = SAVE(J,I)*R1                                               GEAR5940
            H = HOLD*RACUM                                                      GEAR5950
            DO 770 I=1,N                                                        GEAR5960
      770   Y(1,I) = SAVE(1,I)                                                  GEAR5970
            IDOUB = K                                                           GEAR5980
            GO TO (130, 250,640),IRET1                                          GEAR5990
      780   KFLAG = -4                                                          GEAR6000
            GO TO 470                                                           GEAR6010
            END                                                                 GEAR6020
```

```
C                                                                       MINV  10
C     ..................................................................MINV  20
C                                                                       MINV  30
C        SUBROUTINE MINV                                                MINV  40
C                                                                       MINV  50
C        PURPOSE                                                        MINV  60
C           INVERT A MATRIX                                             MINV  70
C                                                                       MINV  80
C        USAGE                                                          MINV  90
C           CALL MINV(A,N,D,L,M)                                        MINV 100
C                                                                       MINV 110
C        DESCRIPTION OF PARAMETERS                                      MINV 120
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY  MINV 130
C               RESULTANT INVERSE.                                      MINV 140
C           N - ORDER OF MATRIX A                                       MINV 150
C           D - RESULTANT DETERMINANT                                   MINV 160
C           L - WORK VECTOR OF LENGTH N                                 MINV 170
C           M - WORK VECTOR OF LENGTH N                                 MINV 180
C                                                                       MINV 190
C        REMARKS                                                        MINV 200
C           MATRIX A MUST BE A GENERAL MATRIX                           MINV 210
C                                                                       MINV 220
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MINV 230
C           NONE                                                        MINV 240
C                                                                       MINV 250
C        METHOD                                                         MINV 260
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   MINV 270
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    MINV 280
C           THE MATRIX IS SINGULAR.                                     MINV 290
C                                                                       MINV 300
C     ..................................................................MINV 310
C                                                                       MINV 320
      SUBROUTINE MINV(A,N,D,L,M)                                        MINV 330
      DIMENSION A(1),L(1),M(1)                                          MINV 340
C                                                                       MINV 350
```

```
C          .............................................................MINV 360
C                                                                       MINV 370
C          IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  MINV 380
C          C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION     MINV 390
C          STATEMENT WHICH FOLLOWS.                                       MINV 400
C                                                                       MINV 410
C       DOUBLE PRECISION A,D,BIGA,HOLD                                    MINV 420
C                                                                       MINV 430
C          THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS    MINV 440
C          APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS      MINV 450
C          ROUTINE.                                                       MINV 460
C                                                                       MINV 470
C          THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO      MINV 480
C          CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT  MINV 490
C          10 MUST BE CHANGED TO DABS.                                    MINV 500
C                                                                       MINV 510
C          .............................................................MINV 520
C                                                                       MINV 530
C          SEARCH FOR LARGEST ELEMENT                                     MINV 540
C                                                                       MINV 550
        D=1.0                                                           MINV 560
        NK=-N                                                           MINV 570
        DO 80 K=1,N                                                     MINV 580
        NK=NK+N                                                         MINV 590
        L(K)=K                                                          MINV 600
        M(K)=K                                                          MINV 610
        KK=NK+K                                                         MINV 620
        BIGA=A(KK)                                                      MINV 630
        DO 20 J=K,N                                                     MINV 640
        IZ=N*(J-1)                                                      MINV 650
        DO 20 I=K,N                                                     MINV 660
        IJ=IZ+I                                                         MINV 670
     10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20                             MINV 680
     15 BIGA=A(IJ)                                                      MINV 690
        L(K)=I                                                          MINV 700
        M(K)=J                                                          MINV 710
```

```
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
```

```
   55 CONTINUE                                                          MINV1080
C                                                                       MINV1090
C        REDUCE MATRIX                                                  MINV1100
C                                                                       MINV1110
      DO 65 I=1,N                                                       MINV1120
      IK=NK+I                                                           MINV1130
      HOLD=A(IK)                                                        MINV1140
      IJ=I-N                                                            MINV1150
      DO 65 J=1,N                                                       MINV1160
      IJ=IJ+N                                                           MINV1170
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
   65 CONTINUE                                                          MINV1220
C                                                                       MINV1230
C        DIVIDE ROW BY PIVOT                                            MINV1240
C                                                                       MINV1250
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
C                                                                       MINV1320
C        PRODUCT OF PIVOTS                                              MINV1330
C                                                                       MINV1340
      D=D*BIGA                                                          MINV1350
C                                                                       MINV1360
C        REPLACE PIVOT BY RECIPROCAL                                    MINV1370
C                                                                       MINV1380
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
C                                                                       MINV1410
C        FINAL ROW AND COLUMN INTERCHANGE                               MINV1420
C                                                                       MINV1430
```

```
      K=N                                                               MINV1440
  100 K=(K-1)                                                           MINV1450
      IF(K) 150,150,105                                                 MINV1460
  105 I=L(K)                                                            MINV1470
      IF(I-K) 120,120,108                                               MINV1480
  108 JQ=N*(K-1)                                                        MINV1490
      JR=N*(I-1)                                                        MINV1500
      DO 110 J=1,N                                                      MINV1510
      JK=JQ+J                                                           MINV1520
      HOLD=A(JK)                                                        MINV1530
      JI=JR+J                                                           MINV1540
      A(JK)=-A(JI)                                                      MINV1550
  110 A(JI) =HOLD                                                       MINV1560
  120 J=M(K)                                                            MINV1570
      IF(J-K) 100,100,125                                               MINV1580
  125 KI=K-N                                                            MINV1590
      DO 130 I=1,N                                                      MINV1600
      KI=KI+N                                                           MINV1610
      HOLD=A(KI)                                                        MINV1620
      JI=KI-K+J                                                         MINV1630
      A(KI)=-A(JI)                                                      MINV1640
  130 A(JI) =HOLD                                                       MINV1650
      GO TO 100                                                         MINV1660
  150 RETURN                                                            MINV1670
      END                                                               MINV1680
```

```
      SUBROUTINE DRGERT(DIFFUN,PEDERV,OUTP,VALINT)                      DRGE  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         DRGE  20
C***********************************************************************DRGE  30
C*        THIS SYSTEM WAS DESIGNED HAVING IN MIND THE FOLLOWING        *DRGE  40
C*             OBJECTIVES :                                             DRGE  50
C*                                                                     *DRGE  60
C*        1          TO MAKE AS EASY AS POSSIBLE THE USE OF THE NUME-  *DRGE  70
C*                   RICAL INTEGRATION PROGRAM WRITTEN BY C. GEAR.     *DRGE  80
C*                                                                     *DRGE  90
C*        2          TO TEST THE  CRITERION FOR RESTRICTION OF THE     *DRGE 100
C*                   STEP SIZE BECAUSE OF  ACCURACY REQUIREMENTS       *DRGE 110
C*                   AND THEREFORE DETERMINE IF THE SYSTEM BEING SOLVED *DRGE 120
C*                   IS A STIFFF ONE. THIS WILL PERMIT SELECTION OF THE *DRGE 130
C*                   ALGORITHM TO SOLVE  THE PROBLEM IN QUESTION       *DRGE 140
C*                   EFFICIENTLY                                        DRGE 150
C*                                                                     *DRGE 160
C*        3          TO GIVE THE USER THE    OPTION   OF CALCULATING   *DRGE 170
C*                   THE VALUES OF THE DEPENDENT VARIABLES AT CERTAIN  *DRGE 180
C*                   SPECIFIC VALUES    OF THE INDEPENDNT VARIABLE     *DRGE 190
C*                   (EXACT INTERVALS).                                 DRGE 200
C*                                                                     *DRGE 210
C*        4          TO ALLOW USER TO PROVIDE THE INITIAL VALUES OF    *DRGE 220
C*                   THE DEPENDENT VARIABLES BY CARD OR BY  A SUBROUTINE *DRGE 230
C*                                                                     *DRGE 240
      CONTINUE                                                          DRGE 250
C*        5          TO ALLOW FOR STANDARD OUTPUT OR SPECIAL OUTPUT    *DRGE 260
C*                   ROUTINE IF SO DESIRED BY USER.                    *DRGE 270
C*                                                                     *DRGE 280
C*        6          TO PROVIDE FOR PREESPECIFICATION OF PARTICULAR    *DRGE 290
C*                   OPTION OF SOLUTION BY USER.                       *DRGE 300
C*                                                                     *DRGE 310
C*        7          TO ALLOW FOR SUPRESSING OF PRINTED OUTPUT OF PARA *DRGE 320
C*                   METER LIST.                                        DRGE 330
C*                                                                     *DRGE 340
      CONTINUE                                                          DRGE 350
```

```
C*                                                                      *DRGE 360
C*        TO CALL THE SYSTEM USE :                                      *DRGE 370
C*     CALL DRGERT(DIFFUN,PEDERV,OUTP,VALINT)                           *DRGE 380
C*                                                                      *DRGE 390
C*        WHERE:                                                        *DRGE 400
C*                                                                      *DRGE 410
C*             DIFFUN    NAME OF THE SUBROUTINE TO CALCULATE THE RIGHT  *DRGE 420
C*                       HAND SIDE OF THE DIFFERENTIAL EQUATIONS        *DRGE 430
C*                                                                      *DRGE 440
C*             PEDERV    NAME OF THE SUBROUTINE TO CALCULATE THE        *DRGE 450
C*                       JACOBIAN.                                      *DRGE 460
C*                                                                      *DRGE 470
C*             OUTP      NAME OF THE SUBROUTINE FOR PRINTED OUTPUT      *DRGE 480
C*                                                                      *DRGE 490
C*             VALINT    SUBROUTINE TO PROVIDE INITIAL VALUES OF TH     *DRGE 500
C*                       DEPENDENT VARIABLES.                            DRGE 510
       CONTINUE                                                          DRGE 520
C*                                                                      *DRGE 530
C*                       THE LAST THREE SUBROUTINES ARE OPTIONAL AND    *DRGE 540
C*                       SHOULD BE INCLUDED ONLY IF THEY ARE            *DRGE 550
C*                       REQUIRED BY THE USER.                           DRGE 560
C*                       HOWEVER MUST INCLUDE THE NAMES IN              *DRGE 570
C*                       CALL INSTRUCTION.                              *DRGE 580
C*                                                                      *DRGE 590
C***********************************************************************DRGE 600
       EXTERNAL DIFFUN,PEDERV,OUTP,VALINT                                DRGE 610
       COMMON /BLK1/J,JSTART,JO,JF,MAXDER,MF,N,IE,IF,IC,IM,KFLAG         DRGE 620
       COMMON/BLK2/H,HMIN,SFL,EPS,FEPS,T,SPLM,SPLM1,DELTA,FSEL,HMAX,T1   DRGE 630
       COMMON/BLK4/ SAVE(12,25),Y(8,25),YMAX(25),ERROR(25),YF(25)        DRGE 640
       COMMON/BLK5/ PW(625),PPW(625)                                     DRGE 650
       COMMON/BLK7/ LLL(25),MMM(25)                                      DRGE 660
       COMMON /BLK15/ EPS1,EPS2                                          DRGE 670
       COMMON/BLK17/ JORDER                                              DRGE 680
       COMMON /BLK20/ IP                                                 DRGE 690
       COMMON NOFNS                                                      DRGE 700
       CALL ERRSET(208,999,-1,1)                                         DRGE 710
```

```
      CALL ERRSET(209,999,-1,1)                                         DRCE 720
      CALL DATARD                                                       DRCE 730
      CALL INVAR                                                        DRCE 740
      CALL PARESC                                                       DRCE 750
C***********************************************************************DRCE 760
C*    IF THE USER HAS DECIDED WHAT METHOD TO USE READ 'MF' FROM        *DRCE 770
C*      THE NEXT CARD.                                                 *DRCE 780
C*    MF         METHOD FLAG  FOR THE INTEGATION METHOD TO BE USED     *DRCE 790
C*                  0                ADAM'S PREDICTOR CORRECTOR        *DRCE 800
C*                  1                GEAR'S STIFF OPTION PROVIDING  'PEDERV'  *DRCE 810
C*                  2                GEAR'S STIFF METHOD WITHOUT 'PEDERV'     *DRCE 820
C-----------------------------------------------------------------------DRCE 830
C                                                                       DRCE 840
      IF(IM .NE. 0) GO TO 396                                           DRCE 850
      READ(5,1004)MF                                                    DRCE 860
 1004 FORMAT(I1)                                                        DRCE 870
      IE = IF                                                           DRCE 880
      SPLM1 = SPLM                                                      DRCE 890
      EPS = EPS1                                                        DRCE 900
      WRITE(6,570)MF                                                    DRCE 910
  570 FORMAT(////1X,37('*')/                                            DRCE 920
     |          1X,'*METHOD FLAG SELECTED BY USER IS ',1X,I2,'*'/       DRCE 930
     |  1X,37('*'))                                                     DRCE 940
      CALL ESCINT                                                       DRCE 950
      RETURN                                                            DRCE 960
C                                                                       DRCE 970
C***********************************************************************DRCE 980
C*    TRY ADAMS METHOD WITH EPS = 1.D-07                               *DRCE 990
C-----------------------------------------------------------------------DRCE1000
C                                                                       DRCE1010
  396 IE = 0                                                            DRCE1020
      SPLM1 = 0.01D00*SPLM                                              DRCE1030
      IF(SPLM1 .LT. 0.5D00) SPLM1 = 0.5D00                              DRCE1040
      IF(SPLM1 .GT. 1.0D00) SPLM1 = 1.0D00                              DRCE1050
      MF = 0                                                            DRCE1060
      EPS = 1.0D-07                                                     DRCE1070
```

```
      CALL ESCINT                                                       CRGE1080
      JC = J                                                            CRGE1090
      H1A = T/JC                                                        CRGE1100
      WRITE(6,315)EPS,H1A                                               CRGE1110
  315 FORMAT(///1X,'FOR EPS = ',1X,D15.7/ ' ADAM''S PREDICTOR CORRECTOR CRGE1120
     |AVERAGE STEP SIZE IS ',D15.7)                                     CRGE1130
      WRITE(6,316)JORDER                                                CRGE1140
  316 FORMAT(/1X,'THE ORDER OF THE INTEGRATION IS ',I2/)                CRGE1150
      JORD1 = JORDER                                                    CRGE1160
C***********************************************************************CRGE1170
C*    TRY ADAMS METHOD WITH EPS = 1.D-04                               *CRGE1180
C                                                                       CRGE1190
C-----------------------------------------------------------------------CRGE1200
      CALL INVAR                                                        CRGE1210
      EPS = 1.0D-04                                                     CRGE1220
      CALL ESCINT                                                       CRGE1230
      JF = J                                                            CRGE1240
      H2A = T/JF                                                        CRGE1250
      WRITE(6,315)EPS,H2A                                               CRGE1260
      WRITE(6,316)JORDER                                                CRGE1270
      JORD2 = JORDER                                                    CRGE1280
      F1 = (1.D-04)**(1.D00/DFLOAT(JORD2))                              CRGE1290
      F2 = (1.D-07)**(1.D00/DFLOAT(JORD1))                              CRGE1300
      H2C = H1A*(F1/F2)                                                 CRGE1310
      WRITE(6,317)EPS,H2C                                               CRGE1320
  317 FORMAT(1X,'STEP SIZE CALCULATED FOR EPS = ',D15.7,3X,'IS =',D15.7)CRGE1330
      IF (H2A .LT. 0.9D00*H2C) MF = IM                                  CRGE1340
      IF (H2A .LT. 0.9D00*H2C) WRITE(6,318)MF                           CRGE1350
  318 FORMAT(///1X,44('*')/                                             CRGE1360
     |           1X,'*METHOD SELECTED IS GEAR''S STIFF OPTION   ',I1,'*'/ CRGE1370
     | 1X,44('*'))                                                      CRGE1380
      IF (H2A .GE. 0.9D00*H2C) MF = 0                                   CRGE1390
      IF( H2A .GE. 0.9D00*H2C) WRITE(6,321)                             CRGE1400
  321 FORMAT(///,1X,47('*')/                                            CRGE1410
     |          1X,'*METHOD SELECTED IS ADAM''S PREDICTOR CORRECTOR*'/    CRGE1420
     | 1X,47('*'))                                                      CRGE1430
```

```
      EPS = EPS1                                               CRGE1440
      SPLM1 = SPLM                                             CRGE1450
      IE = IF                                                  CRGE1460
      CALL INVAR                                               CRGE1470
      CALL ESCINT                                              CRGE1480
      RETURN                                                   CRGE1490
  852 MF = 0                                                   CRGE1500
      CALL INVAR                                               CRGE1510
      CALL ESCINT                                              CRGE1520
      RETURN                                                   CRGE1530
      END                                                      CRGE1540
```

```
      SUBROUTINE DATARD                                                 DATA  10
      IMPLICIT REAL*8 (A-H,Q-Z)                                         DATA  20
      IMPLICIT INTEGER*4 (O)                                            DATA  30
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT                                DATA  40
      COMMON /BLK1/J,JSTART,JO,JF,MAXDER,MF,N,IE,IF,IO,IM,KFLAG         DATA  50
      COMMON/BLK2/H,HMIN,SEL,EPS,FEPS,T,SPLM,SPLM1,DELTA,FSEL,HMAX,T1   DATA  60
      COMMON/BLK4/ SAVE(12,25),Y(8,25),YMAX(25),ERROR(25),YF(25)        DATA  70
      COMMON/BLK5/ PW(625),PPW(625)                                     DATA  80
      COMMON/BLK7/ LLL(25),MMM(25)                                      DATA  90
      COMMON/BLK8/ OHEAD(20)                                            DATA 100
      COMMON /BLK9/Y1(25)                                               DATA 110
      COMMON /BLK15/ EPS1,EPS2                                          DATA 120
      COMMON /BLK20/ IP                                                 DATA 130
      COMMON NCFNS                                                      DATA 140
C***********************************************************************DATA 150
C*    ----------    READ THE TITLE CARD     ------------               *DATA 160
C-----------------------------------------------------------------------DATA 170
C                                                                       DATA 180
 3535 READ(5,1005)(OHEAD(I),I=1,20)                                     DATA 190
 1005 FORMAT(20A4)                                                      DATA 200
C                                                                       DATA 210
C***********************************************************************DATA 220
C*             READ THE FOLLOWING PARAMETRS                            *DATA 230
C*        IE      0      IT IS NOT REQUIRED OUTPUT AT EXACT INTERVALS  *DATA 240
C*                1      IT IS REQUIRED OUTPUT AT EXACT INTERVALS      *DATA 250
C*        IO      0      IT IS DESIRED STANDARD OUTPUT                 *DATA 260
C*                1      USER IS TO INCLUDE HIS OWN OUTPUT SUBROUTINE  *DATA 270
C*        IV      0      INITIAL VALUES PROVIDED  THROUGH PUNCHED CARDS *DATA 280
C*                1      INITIAL VALUES WILL BE PROVIDED BY SUBROUTINE *DATA 290
C*                       'VALINT'.                                     *DATA 300
C*        IM      0      THE USER DECIDES WHICH METHOD TO USE          *DATA 310
C*                1      THE USER WANTS THIS SYSTEM TO DECIDE BETWEEN  *DATA 320
C*                       ADM'S PREDICTOR CORRECTOR  AND GEAR'S STIFF   *DATA 330
C*                       METHOD ( 'PEDERV' MUST BE PROVIDED)            DATA 340
C*                2      THE USER WANTS THIS SUBROUTINE TO DECIDE      *DATA 350
```

```
C*                           BETWEEN ADAMS P.C. AND GEAR'S METHODS OPTION 2  *DATA 360
C*                           (IN THIS CASE THE JACOBIAN IS APPROXIMATED NU-  *DATA 370
C*                           MERICALLY BY GEAR'S PROGRAM.                     *DATA 380
C*    IP  0      THE USER WANTS THE PARAMETERS TO BE PRINTED                  *DATA 390
C*        1      THE USER WANTS TO BE SUPPRESSED THE PRINTING OF THE          *DATA 400
C*               INTEGRATION PARAMETERS                                       *DATA 410
C-----------------------------------------------------------------------------DATA 420
C                                                                              DATA 430
      READ(5,1003)IF,IO,IV,IM,IP                                               DATA 440
 1003 FORMAT(5I1)                                                              DATA 450
C                                                                              DATA 460
C                                                                              DATA 470
C                                                                              DATA 480
C*****************************************************************************DATA 490
C*    READ THE FOLLOWING PARAMETERS:                                          *DATA 500
C*    SEL        INTERVAL IN WHICH IT IT DESIRED OUTPUT                       *DATA 510
C*    N         THE NUMBER OF EQUATIONS                                       *DATA 520
C*    EPS        THE ERROR ALLOWANCE                                          *DATA 530
C*    T         THE INITAL VALUE OF THE INDEPENDENT VARIABLE                  *DATA 540
C*    SPLM      THE SUPERIOR LIMIT OF THE NUMERICAL INTEGRATION               *DATA 550
C-----------------------------------------------------------------------------DATA 560
C                                                                              DATA 570
C                                                                              DATA 580
      READ(5,1000)SEL,N,EPS1,T1,SPLM                                           DATA 590
 1000 FORMAT(D6.2,  3X,I2 , 3(3X,D6.0))                                        DATA 600
C                                                                              DATA 610
C                                                                              DATA 620
C*****************************************************************************DATA 630
C*  TEST IF THE INITIAL VALUES OF THE DEPENDENT VARIABLE WILL BE READ        *DATA 640
C*FROM CARDS OR  WILL BE CALCULATED BY A SUBROUTINE                           *DATA 650
C-----------------------------------------------------------------------------DATA 660
C                                                                              DATA 670
      IF ( IV .EQ. 0) GO TO 2510                                               DATA 680
C                                                                              DATA 690
      CALL VALINT(Y1,N)                                                        DATA 700
      GO TO 2511                                                               DATA 710
```

```
C                                                                       DATA 720
C***********************************************************************DATA 730
C*  READ THE INITIAL VALUES OF THE DEPENDENT VARIABLES                 *DATA 740
C*       IF THE USER IS GOING TO PROVIDE THE INITIAL VALUES BY CARD    *DATA 750
C-----------------------------------------------------------------------DATA 760
C                                                                       DATA 770
 2510 CONTINUE                                                          DATA 780
      READ(5,1010) ( Y1(I), I=1,N)                                      DATA 790
 1010 FORMAT(5(D12.4,4X))                                               DATA 800
C                                                                       DATA 810
2511  RETURN                                                            DATA 820
      END                                                               DATA 830
```

```
      SUBROUTINE PARESC                                                 PARE  10
      IMPLICIT REAL*8 (A-H,Q-Z)                                         PARE  20
      IMPLICIT INTEGER*4 (C)                                            PARE  30
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT                                PARE  40
      COMMON /BLK1/J,JSTART,JC,JF,MAXDER,MF,N,IE,IF,IC,IM,KFLAG         PARE  50
      COMMON/BLK2/H,HMIN,SEL,EPS,FEPS,T,SPLM,SPLM1,DELTA,FSEL,HMAX,T1   PARE  60
      COMMON/BLK4/ SAVE(12,25),Y(8,25),YMAX(25),ERROR(25),YF(25)        PARE  70
      COMMON/BLK5/ PW(625),PPW(625)                                     PARE  80
      COMMON/BLK7/ LLL(25),MMM(25)                                      PARE  90
      COMMON/BLK8/ CHEAD(20)                                            PARE 100
      COMMON /BLK15/ EPS1,EPS2                                          PARE 110
      COMMON /BLK20/ IP                                                 PARE 120
      COMMON NOFNS                                                      PARE 130
C                                                                       PARE 140
      IF (IP .EQ. 1 )                                                   PARE 150
     |                          RETURN                                  PARE 160
C***********************************************************************PARE 170
C*     WRITE ALL THE PARAMETERS TO THE NUMERICAL INTEGRATION           *PARE 180
C-----------------------------------------------------------------------PARE 190
C                                                                       PARE 200
C                                                                       PARE 210
      WRITE(6,1020)CHEAD                                                PARE 220
 1020 FORMAT(1H1/////1X,20A4)                                           PARE 230
      WRITE(6,1030)SEL,    N,H,HMIN,EPS1,MAXDER,T                       PARE 240
 1030 FORMAT(1H-,'OUTPUT AT INTERVALS OF ',D9.2/                        PARE 250
     1' NUMBER OF EQUATIONS IS ',I2/' INITIAL STEP SIZE IS ',D9.2/      PARE 260
     2' MINIMUM STEP SIZE IS ',D9.2/' TOLERANCE IS ',D9.2/              PARE 270
     3' MAXIMUM DERIVATIVE TO BE USED IS ',I1/' INITIAL VALUE OF THE INDPARE 280
     4EPENDENT VARIABLE IS ' ,D9.2)                                     PARE 290
      WRITE(6,1041)HMAX,SPLM                                            PARE 300
 1041 FORMAT(1X,'MAXIMUM STEP SIZE ALLOWED IS ',D15.6/                  PARE 310
     1' SUPERIOR LIMIT OF THIS INTEGRATION IS ',D15.6)                  PARE 320
      IF(IF .EQ. 1 )                                                    PARE 330
     |                  WRITE(6,520)                                    PARE 340
  520 FORMAT(1X,'EXACT INTERVALS WILL BE PROVIDED')                     PARE 350
```

```
      IF ( IO .EQ. 1 )                                                  PARE 360
     |                     WRITE(6,525)                                 PARE 370
  525 FORMAT(1X,'THE USER IS GOING TO INCLUDE HIS OUTPUT SUBROUTINE')   PARE 380
      IF(IV .EQ. 1 )                                                    PARE 390
     |                WRITE(6,530)                                      PARE 400
  530 FORMAT(1X,'THE INITIAL VALUES WILL BE PROVIDED BY THE SUBROUTINE  PARE 410
     '''VALINT'' ')                                                     PARE 420
      IF(IM .EQ. 0)                                                     PARE 430
     |              WRITE(6,540)                                        PARE 440
  540 FORMAT(1X,'THE USER HAS SELECTED THE INTEGRATION METHOD')         PARE 450
      IF( IM .EQ. 1)                                                    PARE 460
     |                WRITE(6,550)                                      PARE 470
  550 FORMAT(1X,'ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR''S STIFFPARE 480
     | METHOD WILL BE SELECTED')                                        PARE 490
      IF (IM .EQ. 2)                                                    PARE 500
     |              WRITE(6,560)                                        PARE 510
  560 FORMAT(1X,'ADAM''S PREDICTOR CORRECTOR OR OPTION 2 OF GEAR''S STIFPARE 520
     'F METHOD WILL BE SELECTED')                                       PARE 530
      WRITE(6,1040) (Y(1,I), I = 1,N)                                   PARE 540
 1040 FORMAT(1H0,'THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE'/   PARE 550
     110(2X,D12.4)/////)                                                PARE 560
      LIN = 60                                                          PARE 570
      RETURN                                                            PARE 580
      END                                                               PARE 590
```

```
      SUBROUTINE INVAR                                                  INVA  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         INVA  20
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT                                INVA  30
      COMMON /BLK1/J,JSTART,JC,JF,MAXDER,MF,N,IE,IF,IC,IM,KFLAG         INVA  40
      COMMON/BLK2/H,HMIN,SEL,EPS,FEPS,T,SPLM,SPLM1,DELTA,FSEL,HMAX,T1   INVA  50
      COMMON/BLK4/ SAVE(12,25),Y(8,25),YMAX(25),ERROR(25),YF(25)        INVA  60
      COMMON/BLK5/ PW(625),PPW(625)                                     INVA  70
      COMMON/BLK7/ LLL(25),MMM(25)                                      INVA  80
      COMMON /BLK9/Y1(25)                                               INVA  90
      COMMON /BLK10/ LIN                                                INVA 100
      COMMON /BLK15/ EPS1,EPS2                                          INVA 110
      COMMON NOFNS                                                      INVA 120
C                                                                       INVA 130
C                                                                       INVA 140
C***********************************************************************INVA 150
C*                INITIALIZE ALL THE VARIABLES REQUIRED BY THE NUMERICAL *INVA 160
C*                INTEGRATION.                                          *INVA 170
C*          NOFNS      ACCUMULATE THE NUMBER OF FUNCTION EVALUATIONS    *INVA 180
C*          J          NUMBER OF STEPS TAKEN                            *INVA 190
C*          H          STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP       *INVA 200
C*          HMIN       THE MINIMUM STEP SIZE THAT CAN BE TAKEN IN THE   *INVA 210
C*                     NUMERICAL INTEGRATION.                            INVA 220
C*          MAXDER     THE MAXIMUM ORDER OF THE INTEGRATION             *INVA 230
C*          JSTART     REFER TO  GEAR'S PROGRAM COMMENTS FOR EXPLANATION *INVA 240
C*          YMAX       ARRAY USED BY GEAR METHOD MUST BE ASSIGNED VALUE *INVA 250
C*                     OF UNITY TO EACH OF ITS ELEMENTS BEFORE STARTING *INVA 260
C*                     THE NUMERICAL INTEGRATION.                       *INVA 270
C-----------------------------------------------------------------------INVA 280
C                                                                       INVA 290
      T = T1                                                            INVA 300
      NOFNS = 0                                                         INVA 310
      J = 0                                                             INVA 320
      JSTART = 0                                                        INVA 330
      H = 1.D-04                                                        INVA 340
      LIN = 60                                                          INVA 350
```

```
      HMIN = 1.0D-10                                                    INVA 360
      MAXDER = 6                                                        INVA 370
C                                                                       INVA 380
C                                                                       INVA 390
      DO 55 I = 1,N                                                     INVA 400
      YMAX(I) = 1.D00                                                   INVA 410
   55 CONTINUE                                                          INVA 420
C                                                                       INVA 430
C                                                                       INVA 440
C***********************************************************************INVA 450
C*SET OTHER VARIABLES TO PROPER VALUES  BEFORE STRATING THE INTEGRATION*INVA 460
C*    HMAX   THE MAXIMUM STEP SIZE THAT CAN BE TAKEN IN THE INTEGRATION*INVA 470
C*    FEPS  IS THE MAXIMUM ERROR ALLOWANCE FOR THE SYSTEM THAT         *INVA 480
C*  IS BEING INTEGRATED.   IS THREE ORDERS OF MAGNITUDE LARGER THAN EPS*INVA 490
C-----------------------------------------------------------------------INVA 500
C                                                                       INVA 510
      HMAX = SPLM                                                       INVA 520
C***********************************************************************INVA 530
C*  A VALUE EQUAL TO THE INTERVAL IN QUESTION                          *INVA 540
C* IF EXACT INTERVAL ARE REQUIRED THE MAXIMUM STEP SIZE IS ASSIGNED    *INVA 550
C-----------------------------------------------------------------------INVA 560
      IF (IF .EQ. 1) HMAX = SEL                                         INVA 570
C                                                                       INVA 580
C                                                                       INVA 590
      FEPS = EPS1*1.D+03                                                INVA 600
      DELTA = SEL                                                       INVA 610
      FSEL = SEL                                                        INVA 620
C                                                                       INVA 630
      DO 60 I = 1,N                                                     INVA 640
   60 Y(1,I) = Y1(I)                                                    INVA 650
      RETURN                                                            INVA 660
      END                                                               INVA 670
```

```
      SUBROUTINE ESCINT                                                 ESCI  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         ESCI  20
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT                                ESCI  30
      COMMON /BLK1/J,JSTART,JO,JF,MAXDER,MF,N,IE,IF,IO,IM,KFLAG         ESCI  40
      COMMON/BLK2/H,HMIN,SEL,EPS,FEPS,T,SPLM,SPLM1,DELTA,FSEL,HMAX,T1   ESCI  50
      COMMON/BLK4/ SAVE(12,25),Y(8,25),YMAX(25),ERROR(25),YF(25)        ESCI  60
      COMMON/BLK5/ PW(625),PPW(625)                                     ESCI  70
      COMMON/BLK7/ LLL(25),MMM(25)                                      ESCI  80
      COMMON /BLK10/ LIN                                                ESCI  90
      COMMON/BLK17/ JORDER                                              ESCI 100
      COMMON NOFNS                                                      ESCI 110
C***********************************************************************ESCI 120
C*  INCREASE BY ONE THE NUMBER OF STEPS TAKEN BEFORE PERFORMING        *ESCI 130
C*     THE NEXT ONE.                                                   *ESCI 140
C-----------------------------------------------------------------------ESCI 150
C                                                                       ESCI 160
      JSWIE = 3                                                         ESCI 170
      KFLAG = 1                                                         ESCI 180
   80 J = J + 1                                                         ESCI 190
      JSWIE = JSWIE + 1                                                 ESCI 200
      IF (KFLAG .GT. 0) TC = T                                          ESCI 210
C                                                                       ESCI 220
C***********************************************************************ESCI 230
C* 'TC' CONTAINS THE INDEP. VARIABLE VALUE BEFORE INTEGRATION STEP      ESCI 240
C-----------------------------------------------------------------------ESCI 250
C                                                                       ESCI 260
      IF(IE .EQ. 1 ) TC = T                                             ESCI 270
      IF(JSWIE .EQ. 1 ) GO TO 315                                       ESCI 280
      IF(JSWIE .NE. 2 ) GO TO 315                                       ESCI 290
      H = DT                                                            ESCI 300
C***********************************************************************ESCI 310
C* PERFORM ONE INTEGRATION STEP                                        *ESCI 320
C-----------------------------------------------------------------------ESCI 330
C                                                                       ESCI 340
C                                                                       ESCI 350
```

```
  315 CONTINUE                                                          ESCI 360
      CALL GEARMF(N,T,Y,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG,JSTART,ESCI 37C
     2            MAXDER,PW,PPW,LLL,MMM,DIFFUN,PEDERV)                  ESCI 38C
C                                                                       ESCI 390
C                                                                       ESCI 4CC
C***********************************************************************ESCI 41C
C*      TEST IF THE PREVICUS STEP WAS SUCCESFUL                        *ESCI 42C
C*                                                                     *ESCI 430
C-----------------------------------------------------------------------ESCI 44C
C                                                                       ESCI 45C
      IF (KFLAG .LT. 0 ) GO TO 4CCO                                     ESCI 460
C***********************************************************************ESCI 470
C*  TN CONTAINS THE VALUE OF THE INDEPENDENT VARIABLE AFTER THE INTEGRA*ESCI 48C
C*  TION STEP HAS BEEN EXECUTED.                                       *ESCI 490
C-----------------------------------------------------------------------ESCI 5C0
      IF (IE .EC. 1) TN = T                                             ESCI 51C
C                                                                       ESCI 52C
C                                                                       ESCI 530
C                                                                       ESCI 54C
C***********************************************************************ESCI 550
C* TEST IF THE EXACT INTERVAL FEATURE HAS BEEN REQUESTEC                *ESCI 560
C* IF NCT SKIP THE CORRESPONDING INSTRUCTIONS                           *ESCI 57C
C-----------------------------------------------------------------------ESCI 58C
C                                                                       ESCI 590
      IF ( IE .EC. 0 ) GO TO 4CCO                                       ESCI 6CC
C                                                                       ESCI 61C
C***********************************************************************ESCI 62C
C* IF THE NEXT EXACT POINT HAS BEEN EXCEEDED  DETERMINE THE EXCESS      *ESCI 63C
C* AND REPEAT THE PREVICUS INTEGRATION STEP WITH A NEW  STEP SIZE SUCH *ESCI 64C
C* THAT THE REPEATED STEP WILL GIVE THE VALUE OF THE INDEPENDENT VARIA *ESCI 650
C* BLE EXACTLY AT THE POINT DESIRED                                     *ESCI 660
C-----------------------------------------------------------------------ESCI 67C
C                                                                       ESCI 68C
      IF(T .LT. DELTA*0.9999DCC)GO TO 4CCC                              ESCI 690
      DT = TN - TO                                                      ESCI 7CC
      H = DELTA - TC                                                    ESCI 71C
```

```
      DELTA = DELTA + SEL                                               ESCI 720
      IF(DABS(H) .LT. 1.D-06) GO TO 4000                                ESCI 730
      JSTART = -1                                                       ESCI 740
      JSWIE = 0                                                         ESCI 750
      GO TO 80                                                          ESCI 760
C***********************************************************************ESCI 770
C* DETERMINATION OF THE PROBLEM IN THE INTEGRATION STEP AND CORRECTIONS*ESCI 780
C-----------------------------------------------------------------------ESCI 790
 4000 CONTINUE                                                          ESCI 800
      IF(KFLAG.GT.0) GO TO 81                                           ESCI 810
      IF(KFLAG.EQ. -1) JSTART = -1                                      ESCI 820
      IF(KFLAG.EQ. -2) WRITE(6,225)                                     ESCI 830
  225 FORMAT(1H-,'THE MAXIMUM ORDER SPECIFIED IS TOO LARGE')            ESCI 840
      IF(KFLAG .EQ. -3)   WRITE(6,230)                                  ESCI 850
  230 FORMAT(1H-,'CORRECTOR CONVERGENCE NOT ACHIEVED FOR H.GT.HMIN')    ESCI 860
        IF(KFLAG.EQ. -4) WRITE(6,235)                                   ESCI 870
   235 FORMAT(1H-,'REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED')     ESCI 880
        IF(KFLAG.EQ. -4) GO TO 579                                      ESCI 890
      IF(KFLAG .NE. -1) GO TO 96                                        ESCI 900
      IF(HMIN.LT.1.D-09)  WRITE(6,666)HMIN                              ESCI 910
  666 FORMAT(1H-,12X,'PROGRAM TERMINATED HMIN = ',D15.6)                ESCI 920
      HMIN = HMIN/2.D00                                                 ESCI 930
      GO TO 80                                                          ESCI 940
   81 CONTINUE                                                          ESCI 950
C***********************************************************************ESCI 960
C*     SAVE THE ACTUAL ORDER OF THE INTEGRATION BEFORE  ASSIGNING NEW  *ESCI 970
C*        VALUE TO JSTART                                              *ESCI 980
C-----------------------------------------------------------------------ESCI 990
C                                                                       ESCI1000
      JORDER = JSTART                                                   ESCI1010
      JSTART = 1                                                        ESCI1020
C                                                                       ESCI1030
C***********************************************************************ESCI1040
C* CHECK IF IT IS TIME TO PRODUCE PRINTED OUTPUT                       *ESCI1050
C-----------------------------------------------------------------------ESCI1060
C                                                                       ESCI1070
```

```
      IF(T .LT. FSEL*0.9999D00) GO TO 95                                ESCI1080
      IF (SPLM1 .NE. SPLM) GO TO 95                                     ESCI1090
      FSEL = FSEL + SEL                                                 ESCI1100
C                                                                       ESCI1110
C***********************************************************************ESCI1120
C* TRANSFER THE VALUES OF THE DEPENDENT VARIABLES TO THE OUTPUT VECTOR *ESCI1130
C-----------------------------------------------------------------------ESCI1140
C                                                                       ESCI1150
      DO 222  I = 1,N                                                   ESCI1160
  222 YF(I) = Y(1,I)                                                    ESCI1170
C                                                                       ESCI1180
C***********************************************************************ESCI1190
C* TEST IF THE USER IS GOING TO INCLUDE HIS OWN OUTPUT SUBROUTINE      *ESCI1200
C*IF NOT THE STANDARD UPTUT ROUTINE WILL BE EXECUTED                   *ESCI1210
C-----------------------------------------------------------------------ESCI1220
C                                                                       ESCI1230
      IF ( IO .EQ. 0) GO TO 4100                                        ESCI1240
C***********************************************************************ESCI1250
C*THIS SUBROUTINE WRITTEN BY THE USER ALLOWS HIM TO PRINT ANY OF THE   *ESCI1260
C*FOLLOWING DATA:                                                      *ESCI1270
C*    N        NUMBER OF EQUATIONS                                     *ESCI1280
C*    T        INDEPENDENT VARIABLE (NORMALLY  TIME)                   *ESCI1290
C*    J       NUMBER   OF STEPS TAKEN UP TO THIS POINT IN THE NUM. INT. *ESCI1300
C*    YF      A VECTOR  CONTAINIG THE VALUES OF THE DEPENDENT VARIABLES *ESCI1310
C*            IT HAS 'N' ELEMENTS.                                      ESCI1320
C*    SAVE    TWO DIMENSIONAL ARRAY CONTAINING THE DERIVATIVES IN      *ESCI1330
C*            THE SUBARRAY SAVE(N2,I)) WHERE N2 = N*10 + 1             *ESCI1340
C*            REFER TO GEAR PROGRAM FOR MORE INFORMATION ABOUT THIS    *ESCI1350
C*            PARAMETER                                                 ESCI1360
C*    SEL     TIME INTERVAL FOR PRINTING                               *ESCI1370
C*    DT     PREVIOUS STEP SIZE                                        *ESCI1380
C*    NOFNS  NUMBER OF FUNCTION  EVALUATIONS UP TO THIS POINT          *ESCI1390
C-----------------------------------------------------------------------ESCI1400
C                                                                       ESCI1410
C                                                                       ESCI1420
      CALL OUTP(N,T,J,YF,SAVE,H,DT,NOFNS)                               ESCI1430
```

```
      GO TO 95                                                          ESCI1440
C                                                                       ESCI1450
C                                                                       ESCI1460
C***********************************************************************ESCI1470
C*          STANDARD OUTPUT SECTION                                    *ESCI1480
C-----------------------------------------------------------------------ESCI1490
C                                                                       ESCI1500
 4100 CONTINUE                                                          ESCI1510
      IF(LIN .LE. 56)GO TO 963                                          ESCI1520
   57 WRITE(6,105)                                                      ESCI1530
  105 FORMAT(1H1,3X,'DE. EV. ','ACCUM. STEPS',6X,'STEP SIZE',           ESCI1540
     1'    INDEP. VARIABLE', 2X,'DEP. VAR(1) DEP. VAR(2) ',////)        ESCI1550
      LIN = 4                                                           ESCI1560
 963  WRITE(6,91)NOFNS,J,H ,T                                           ESCI1570
 91   FORMAT(1H ,3X,I6,2X,I6,10X,D15.8,1X,D14.7)                        ESCI1580
      WRITE(6,1950) (YF(I), I= 1,N)                                     ESCI1590
 1950 FORMAT(1H+,58X,4(1X,D15.8,1X),6(/1X,58X,4(1X,D15.8,1X)))          ESCI1600
      LIN = LIN + 1+ N/4                                                ESCI1610
C                                                                       ESCI1620
C***********************************************************************ESCI1630
C* TEST IF THE INTEGRATION HAS REACHED THE SUPERIOR LIMIT              *ESCI1640
C-----------------------------------------------------------------------ESCI1650
C                                                                       ESCI1660
   95 IF ( T.LT.SPLM1) GO TO 80                                         ESCI1670
C                                                                       ESCI1680
      IF(SPLM1 .NE. SPLM) GO TO 96                                      ESCI1690
      WRITE(6,99)NOFNS                                                  ESCI1700
  99  FORMAT(1H0,'TOTAL NUMBER OF FUNCTION EVALUATIONS', I6)            ESCI1710
  96  CONTINUE                                                          ESCI1720
      RETURN                                                            ESCI1730
C                                                                       ESCI1740
C                                                                       ESCI1750
C***********************************************************************ESCI1760
C* IF THE ERROR REQUESTED HAS BEEN SMALLER THAT CAN BE HANDLED         *ESCI1770
C* INCREASE IT UP TO A MAXIMUM OF  1.0E+03 TIMES THE INITIAL ERROR     *ESCI1780
C* USING INCREMENTS OF 1.0E+01 EACH TIME                               *ESCI1790
```

```
C-----------------------------------------------------------------------ESCI1800
C                                                                       ESCI1810
  579 EPS = EPS*10.D00                                                  ESCI1820
      IF (EPS .GT. FEPS) GO TO 581                                      ESCI1830
      WRITE(6,580)EPS                                                   ESCI1840
  580 FORMAT(1X,'NEW TOLERANCE IS' , D15.7)                             ESCI1850
      JSTART = -1                                                       ESCI1860
      GO TO 80                                                          ESCI1870
  581 WRITE(6,582)FEPS                                                  ESCI1880
  582 FORMAT(1X,'ERROR THAT CAN BE HANDLED GREATER THAN ',D15.7/        ESCI1890
     | 1X,'PROGRAM TERMINATED')                                         ESCI1900
 3535 STOP 3535                                                         ESCI1910
 3333 RETURN                                                            ESCI1920
      END                                                               ESCI1930
```

GENERAL PROGRAM LINKAGE

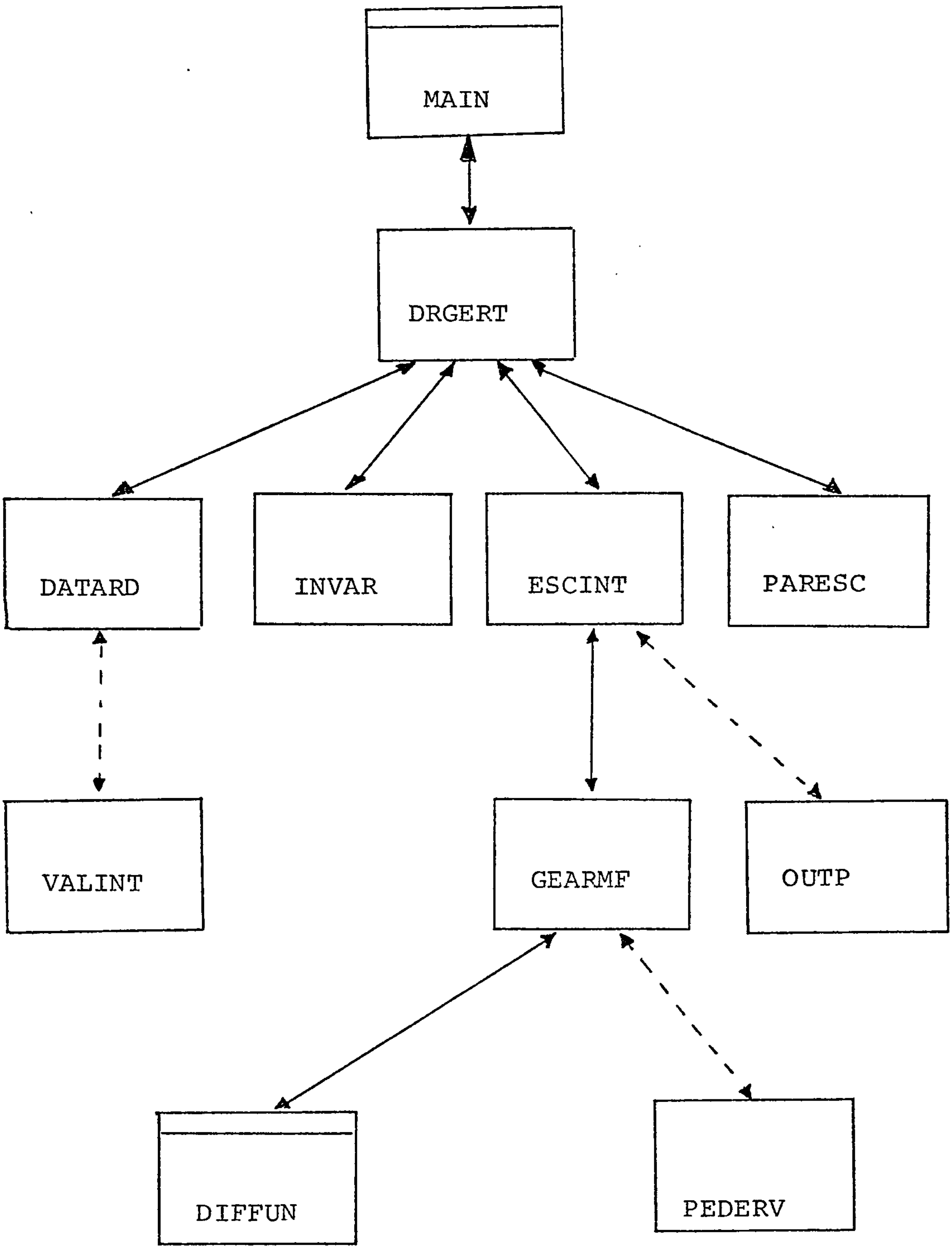


Figure A.1

```
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT
      CALL DRGERT(DIFFUN,PEDERV,OUTP,VALINT)
      STOP
      END
```

```
0001      SUBROUTINE DIFFUN(T,Y,DY)
0002      IMPLICIT REAL*8 (A-H,O-Z)
0003      DIMENSION Y(8,3),DY(3)
0004      COMMON NCFNS
0005      NCFNS = NCFNS + 1
0006      DY(1) = -0.04D00*Y(1,1) + 10000.D00*Y(1,2)*Y(1,3)
0007      DY(2) = 0.04D00*Y(1,1) - 10000.D00*Y(1,2)*Y(1,3)
         1   - 3.D+07*(Y(1,2)**2)
0008      DY(3) = (3.D+07)*(Y(1,2)**2)
0009      RETURN
0010      END
```

```
      SUBROUTINE PEDERV( T,Y,PW,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,3),PW(3,3)
      PW(1,1) = - 0.04
      PW(1,2) =  (1.D+04)*Y(1,3)
      PW(1,3) = (1.D+04)*Y(1,2)
      PW(2,1) =  0.04
      PW(2,2) =  (-1.D+04)*Y(1,3) -(6.D+07)*Y(1,2)
      PW(2,3) = (-1.D+04)*Y(1,2)
      PW(3,1) = 0.0
      PW(3,2) = (6.D+07)*Y(1,2)
      PW(3,3) = 0.0
      RETURN
      END
```

```
PROBLEM 3.1  A SYSTEM OF REACTION RATE EQUATIONS


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  3
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000D 01
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.4000000D 02
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.10000D 01    0.0             0.0



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.3571021D-03

THE ORDER OF THE INTEGRATION IS  1




FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.5208773D-03

THE ORDER OF THE INTEGRATION IS  1

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.3571021D 00


****************************************
*METHOD SELECTED IS GEAR'S STIFF OPTION   1*
****************************************
```

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2)	
54	21	0.46251148D-01	0.10000000D 01	0.96646452D 00	0.30747034D-04	0.33504734D-01
67	28	0.23627886D-01	0.20000000D 01	0.94161388D 00	0.27018458D-04	0.58359104D-01
88	34	0.90820669D-01	0.30000000D 01	0.92190115D 00	0.24382806D-04	0.78074468D-01
103	41	0.59054341D-01	0.40000000D 01	0.90553409D 00	0.22406597D-04	0.94443504D-01
117	48	0.59054341D-01	0.50000000D 01	0.89153130D 00	0.20854088D-04	0.10844784D 00
123	51	0.46108495D 00	0.60000000D 01	0.87928525D 00	0.19596548D-04	0.12069515D 00
129	54	0.46108495D 00	0.70000000D 01	0.86838755D 00	0.18551122D-04	0.13159390D 00
135	57	0.46108495D 00	0.80000000D 01	0.85856108D 00	0.17664847D-04	0.14142125D 00
141	60	0.46108495D 00	0.90000000D 01	0.84960837D 00	0.16901358D-04	0.15037473D 00
147	63	0.46108495D 00	0.10000000D 02	0.84137994D 00	0.16234701D-04	0.15860383D 00
153	66	0.46108495D 00	0.11000000D 02	0.83376227D 00	0.15646007D-04	0.16622208D 00
159	69	0.46108495D 00	0.12000000D 02	0.82666685D 00	0.15121156D-04	0.17331803D 00
164	72	0.46108495D 00	0.13000000D 02	0.82002320D 00	0.14649343D-04	0.17996215D 00
168	75	0.46108495D 00	0.14000000D 02	0.81377438D 00	0.14222160D-04	0.18621140D 00
171	78	0.46108495D 00	0.15000000D 02	0.80787366D 00	0.13832940D-04	0.19211251D 00
174	81	0.46108495D 00	0.16000000D 02	0.80228227D 00	0.13476333D-04	0.19770425D 00
178	84	0.46108495D 00	0.17000000D 02	0.79696768D 00	0.13147986D-04	0.20301917D 00
181	87	0.46108495D 00	0.18000000D 02	0.79190225D 00	0.12844317D-04	0.20808491D 00
184	90	0.46108495D 00	0.19000000D 02	0.78706237D 00	0.12562345D-04	0.21292507D 00
188	93	0.46108495D 00	0.20000000D 02	0.78242775D 00	0.12299580D-04	0.21755995D 00
191	96	0.46108495D 00	0.21000000D 02	0.77798063D 00	0.12053909D-04	0.22200732D 00
194	99	0.46108495D 00	0.22000000D 02	0.77370558D 00	0.11823510D-04	0.22628260D 00
200	102	0.46108495D 00	0.23000000D 02	0.76958915D 00	0.11606864D-04	0.23039925D 00
207	105	0.46108495D 00	0.24000000D 02	0.76561918D 00	0.11402659D-04	0.23436942D 00
212	108	0.46108495D 00	0.25000000D 02	0.76178502D 00	0.11209676D-04	0.23820377D 00
216	111	0.46108495D 00	0.26000000D 02	0.75807713D 00	0.11026949D-04	0.24191185D 00
219	114	0.46108495D 00	0.27000000D 02	0.75448702D 00	0.10853536D-04	0.24550212D 00
222	117	0.46108495D 00	0.28000000D 02	0.75100707D 00	0.10688731D-04	0.24898224D 00
230	120	0.46108495D 00	0.29000000D 02	0.74763025D 00	0.10531744D-04	0.25235922D 00
235	123	0.46108495D 00	0.30000000D 02	0.74435031D 00	0.10382060D-04	0.25563931D 00
243	126	0.46108495D 00	0.31000000D 02	0.74116151D 00	0.10239016D-04	0.25882825D 00
248	129	0.46108495D 00	0.32000000D 02	0.73805867D 00	0.10102239D-04	0.26193123D 00
256	132	0.46108495D 00	0.33000000D 02	0.73503697D 00	0.99711529D-05	0.26495306D 00
261	135	0.46108495D 00	0.34000000D 02	0.73209208D 00	0.98455050D-05	0.26789807D 00
270	138	0.46108495D 00	0.35000000D 02	0.72921992D 00	0.97247662D-05	0.27077035D 00
277	142	0.77830093D-01	0.36000000D 02	0.72641685D 00	0.96087439D-05	0.27357354D 00
281	146	0.77830093D-01	0.37000000D 02	0.72367934D 00	0.94970626D-05	0.27631116D 00
286	150	0.77830093D-01	0.38000000D 02	0.72100438D 00	0.93895009D-05	0.27898623D 00
293	154	0.77830092D-01	0.39000000D 02	0.71838854D 00	0.92857431D-05	0.28160218D 00
300	158	0.77830093D-01	0.40000000D 02	0.71583214D 00	0.91857570D-05	0.28415868D 00

TOTAL NUMBER OF FUNCTION EVALUATIONS   302

```
PROBLEM 3.1  A SYSTEM OF REACTION RATE EQUATIONS


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  3
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.400000D 02
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.400000D 02
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.10000D 01    0.0            0.0



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS    0.3571021D-03

THE ORDER OF THE INTEGRATION IS  1




FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS    0.5208773D-03

THE ORDER OF THE INTEGRATION IS  1

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.3571021D 00


*******************************************
*METHOD SELECTED IS GEAR'S STIFF OPTION   1*
*******************************************
```

DE. EV.	ACCEP. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2)	....
53	20	0.19527442D 00	0.1147023D 01	0.96238189D 00	0.30102399D-04	0.37588006D-01
60	24	0.32136674D 00	0.2054213D 01	0.94043073D 00	0.26852229D-04	0.59542413D-01
69	27	0.32136674D 00	0.3018314D 01	0.92155709D 00	0.24341951D-04	0.78418566D-01
77	31	0.52097727D 00	0.4503391D 01	0.89821670D 00	0.21579976D-04	0.10176172D 00
82	32	0.52097727D 00	0.5024368D 01	0.89119593D 00	0.20818449D-04	0.10878325D 00
87	34	0.52097727D 00	0.6066323D 01	0.87850272D 00	0.19519105D-04	0.12147776D 00
91	36	0.75810386D 00	0.7108277D 01	0.86725504D 00	0.18446315D-04	0.13272651D 00
98	38	0.75810386D 00	0.8624485D 01	0.85285839D 00	0.17173819D-04	0.14712443D 00
100	39	0.75810386D 00	0.9382589D 01	0.84636173D 00	0.16634403D-04	0.15362164D 00
102	40	0.75810386D 00	0.1014069D 02	0.84025410D 00	0.16146007D-04	0.15972975D 00
106	42	0.11266169D 01	0.1165690D 02	0.82902904D 00	0.15293440D-04	0.17095566D 00
108	43	0.11266169D 01	0.1278352D 02	0.82140821D 00	0.14746446D-04	0.17857704D 00
113	44	0.11266169D 01	0.1391013D 02	0.81430273D 00	0.14257675D-04	0.18568302D 00
115	45	0.11266169D 01	0.1503675D 02	0.80764437D 00	0.13818087D-04	0.19234181D 00
118	46	0.11266169D 01	0.1616337D 02	0.80137758D 00	0.13419721D-04	0.19860900D 00
120	47	0.11266169D 01	0.1728998D 02	0.79545639D 00	0.13056452D-04	0.20453055D 00
122	48	0.16794690D 01	0.1841660D 02	0.78984249D 00	0.12723358D-04	0.21014479D 00
124	49	0.16794690D 01	0.2009607D 02	0.78197532D 00	0.12274530D-04	0.21801240D 00
129	50	0.16794690D 01	0.2177554D 02	0.77463314D 00	0.11873037D-04	0.22535499D 00
131	51	0.16794690D 01	0.2345501D 02	0.76774713D 00	0.11511562D-04	0.23224136D 00
134	52	0.16794690D 01	0.2513448D 02	0.76126120D 00	0.11183635D-04	0.23872761D 00
136	53	0.16794690D 01	0.2681395D 02	0.75512889D 00	0.10884288D-04	0.24486023D 00
138	54	0.25033245D 01	0.2849342D 02	0.74931140D 00	0.10609529D-04	0.25067759D 00
140	55	0.25033245D 01	0.3099674D 02	0.74115474D 00	0.10238908D-04	0.25883502D 00
145	56	0.25033245D 01	0.3350006D 02	0.73353746D 00	0.99069087D-05	0.26645263D 00
147	57	0.25033245D 01	0.3600339D 02	0.72638963D 00	0.96076156D-05	0.27360077D 00
150	58	0.25033245D 01	0.3850671D 02	0.71965412D 00	0.93357686D-05	0.28033654D 00
152	59	0.25033245D 01	0.4101004D 02	0.71328355D 00	0.90873311D-05	0.28670736D 00

TOTAL NUMBER OF FUNCTION EVALUATIONS 152

```
      SUBROUTINE DIFFUN(T,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,1),DY(1)
      COMMON NCFNS
      NCFNS = NCFNS + 1
      ZE =    DEXP(50.D00*(.5D00 - (1.D00/Y(1,2))))
      ZZ = -2.D00*(Y(1,2) - 1.75D00) - 30.D00*(Y(1,2) - 1.75D00)*
     1(Y(1,2) - 2.D00) + Y(1,1)*ZE
      DY(1) = (1.D00 - Y(1,1) - Y(1,1)*ZE)
      DY(2) = ZZ
      RETURN
      END
```

```
0001      SUBROUTINE PEDERV( T,Y,PW,M)
0002      IMPLICIT REAL*8 (A-H,O-Z)
0003      DIMENSION Y(8,1),PW(2,2)
0004      PW(1,1) = - 1.D00 - DEXP(50.D00*(0.5D00 - (1.D00/Y(1,2))))
0005      PW(1,2) =-(50.D00*Y(1,1)/Y(1,2)**2)*DEXP(25.D00 - 50.D00/Y(1,2))
0006      PW(2,1) = DEXP(25.D00 - 50.D00/Y(1,2))
0007      PW(2,2) = - 2.D00 - 30.D00*(2*Y(1,2) -    3.75D00)
         1 + (50.D00*Y(1,1)/Y(1,2)**2)*DEXP(25.D00 - 50.D00/Y(1,2) )
0008      RETURN
0009      END
```

THE STIRRED TANK REACTOR PROBLEM

```
OUTPUT AT INTERVALS OF  0.20D 00
NUMBER OF EQUATIONS IS  2
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.2000000D 00
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.1000000D 02
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.10000 01    0.17500 01


FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.1824893D-01

THE ORDER OF THE INTEGRATION IS  5


FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.4117244D-01

THE ORDER OF THE INTEGRATION IS  3

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.2127669D-01


*************************************************
*METHOD SELECTED IS ADAM'S PREDICTOR CORRECTOR*
*************************************************
```

FE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
27	12	0.14952236D-01	0.20000000D 00	0.99452448D 00	0.17606274D 01
44	21	0.94214051D-02	0.40000000D 00	0.98802416D 00	0.17908054D 01
62	29	0.20232696D-02	0.60000000D 00	0.97455641D 00	0.18532563D 01
90	42	0.25904573D-02	0.80000000D 00	0.93411225D 00	0.19333725D 01
114	54	0.13956291D-01	0.10000000D 01	0.83006204D 00	0.20068803D 01
143	67	0.16620666D-03	0.12000000D 01	0.64645693D 00	0.20616815D 01
189	90	0.15234187D-02	0.14000000D 01	0.48736730D 00	0.20582086D 01
229	108	0.66567365D-02	0.16000000D 01	0.44413631D 00	0.20231385D 01
250	120	0.54850084D-01	0.18000000D 01	0.45150044D 00	0.20032042D 01
270	128	0.57791195D-01	0.20000000D 01	0.46875556D 00	0.19960251D 01
282	132	0.52707599D-01	0.22000000D 01	0.48375644D 00	0.19948229D 01
294	136	0.52707599D-01	0.24000000D 01	0.49372520D 00	0.19958876D 01
304	140	0.52707599D-01	0.26000000D 01	0.49908823D 00	0.19974424D 01
314	144	0.52707599D-01	0.28000000D 01	0.50125552D 00	0.19987382D 01
321	148	0.52707599D-01	0.30000000D 01	0.50164324D 00	0.19995701D 01
328	152	0.52707599D-01	0.32000000D 01	0.50128354D 00	0.19999920D 01
336	156	0.52707599D-01	0.34000000D 01	0.50076875D 00	0.20001425D 01
342	160	0.52707599D-01	0.36000000D 01	0.50035548D 00	0.20001520D 01
347	164	0.52707599D-01	0.38000000D 01	0.50010318D 00	0.20001093D 01
353	168	0.52707599D-01	0.40000000D 01	0.49998364D 00	0.20000612D 01
357	172	0.52707599D-01	0.42000000D 01	0.49994679D 00	0.20000258D 01
361	176	0.52707599D-01	0.44000000D 01	0.49994999D 00	0.20000056D 01
365	180	0.52707599D-01	0.46000000D 01	0.49996625D 00	0.19999969D 01
370	184	0.52707599D-01	0.48000000D 01	0.49998216D 00	0.19999947D 01
374	188	0.52707599D-01	0.50000000D 01	0.49999313D 00	0.19999956D 01
378	192	0.52707599D-01	0.52000000D 01	0.49999905D 00	0.19999972D 01
382	196	0.52707599D-01	0.54000000D 01	0.50000139D 00	0.19999986D 01
387	200	0.52707599D-01	0.56000000D 01	0.50000183D 00	0.19999995D 01
391	204	0.52707599D-01	0.58000000D 01	0.50000142D 00	0.20000000D 01
396	208	0.52707599D-01	0.60000000D 01	0.50000083D 00	0.20000002D 01
401	212	0.52707599D-01	0.62000000D 01	0.50000041D 00	0.20000002D 01
405	216	0.52707599D-01	0.64000000D 01	0.50000014D 00	0.20000001D 01
410	220	0.52707599D-01	0.66000000D 01	0.49999996D 00	0.20000001D 01
414	224	0.52707599D-01	0.68000000D 01	0.49999992D 00	0.20000000D 01
419	228	0.52707599D-01	0.70000000D 01	0.49999995D 00	0.20000000D 01
424	232	0.52707599D-01	0.72000000D 01	0.49999993D 00	0.20000000D 01
428	236	0.52707599D-01	0.74000000D 01	0.49999995D 00	0.20000000D 01
434	240	0.52707599D-01	0.76000000D 01	0.49999992D 00	0.20000000D 01
439	244	0.52707599D-01	0.78000000D 01	0.49999995D 00	0.20000000D 01
443	248	0.52707599D-01	0.80000000D 01	0.49999998D 00	0.20000000D 01
447	252	0.52707599D-01	0.82000000D 01	0.49999999D 00	0.20000000D 01
452	256	0.52707599D-01	0.84000000D 01	0.50000000D 00	0.20000000D 01
456	260	0.52707599D-01	0.86000000D 01	0.50000000D 00	0.20000000D 01
461	264	0.52707599D-01	0.88000000D 01	0.50000001D 00	0.20000000D 01
465	268	0.52707599D-01	0.90000000D 01	0.50000001D 00	0.20000000D 01
470	272	0.52707599D-01	0.92000000D 01	0.50000001D 00	0.20000000D 01
475	276	0.52707599D-01	0.94000000D 01	0.50000001D 00	0.20000000D 01
479	280	0.52707599D-01	0.96000000D 01	0.50000001D 00	0.20000000D 01
484	284	0.52707599D-01	0.98000000D 01	0.49999999D 00	0.20000000D 01
488	288	0.52707599D-01	0.10000000D 02	0.49999999D 00	0.20000000D 01

TOTAL NUMBER OF FUNCTION EVALUATIONS 489

```
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BLK1/CA
      DIMENSION CA(3,4,3)
      EXTERNAL DIFFUN,PEDERV,OUTP,VALINT
      DO 5 NRE = 1,3
    5 READ(5,510) (( CA(I,J,NRE), I = 1,3), J=1,4)
  510 FORMAT( 6D10.4)
      WRITE(6,605)
  605 FORMAT(1H1,'COEFFICIENTS'//)
      DO 50 NRE = 1,3
      WRITE(6,610) (( CA(I,J,NRE), I = 1,3), J=1,4)
  610 FORMAT(1H ,3D15.4)
   50 CONTINUE
      CALL DRGERT(DIFFUN,PEDERV,OUTP,VALINT)
      STOP
      END
```

```
      SUBROUTINE DIFFUN(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KC
      REAL*8 LAMBDA
      DIMENSION Y(8,3),DY(3)
      DATA AA/ 1.D00/
      DATA B /25.D00/
      DATA ETADEL/34.42D00/
      DATA KC/0.4D00/
      DATA LAMBDA / 0.D00/
      DATA RE/1.D+7/
      DATA TRPT/6.484D00/
      COMMON NCFNS
      COMMON /COMUN/EPONU,BETA,TERM1,TERM2
      COMMON /COMU2/ ANUM
      COMMON /COMU3/ DELTA
      NCFNS = NCFNS + 1
      BETA = 2.D00*(KC**2)*DSQRT(2.D00*RE)
      PHI = ((((2.D00*RE)**.25)*DSQRT(5.400D00) *ETADEL) - AA)/B
      NCFNS = NCFNS + 1
      DY(1) = Y(1,2)
      DY(2) = Y(1,3)
      Q = PHI*X/ETADEL
      TERM1 = 1.D00 - DEXP(-Q)
      TERM2 = 1.D00 + (Q-1.D00)*DEXP(-Q)
      EPONU = (BETA/2.D00)*(X**2)* Y(1,3)*(TERM1**2)
      CALL DELTAC(X,DELTA)
      IF(X .GT. TRPT) GO TO 25
      ANUM = - Y(1,3)*Y(1,1) + LAMBDA*(1.D00 -  Y(1,2)**2)
     1 - BETA*X* Y(1,3)**2*TERM1*TERM2
      DY(3) = ANUM/(1.D00 + 2.D00*EPONU)
     1          +  DELTA /( 1.D00 + 2.D00*EPONU)
      GO TO 30
   25 DY(3) =((-Y(1,1)*Y(1,3)
     1 + LAMBDA*((1.D00 - Y(1,2))**2))/(1.D00 + EPONU))
     1 + DELTA/( 1.D00 + EPONU)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DELTAC(X,DELTA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CA(3,4,3) ,F(4,3)
      COMMON /BLK1/CA
      DO 2 K = 1,3
      DO 2 J= 2,4
    2 F(J,K) = CA(1,J,K) + CA(2,J,K)* X  + CA(3,J,K) *(X**2)
      TERM1 = (F(3,3) - F(3,1))/(4.D+6)
      TERM2 = (F(2,3) - F(2,1))/(4.D+6)
      DELTA = 2.D00*RE*((F(3,2)*TERM1) - (F(4,2)*TERM2))
      RETURN
      END
```

```
0001            SUBROUTINE PEDERV( X,Y,PW,N)
0002            IMPLICIT REAL*8 (A-H,O-Z)
0003            DIMENSION Y(8,3),PW(3,3)
0004            REAL*8 LAMBDA
0005            DATA LAMBDA / 0.D00/
0006            DATA TRPT/6.484D00/
0007            COMMON /COMUN/EPONU,BETA,TERM1,TERM2
0008            COMMON /COMU2/ ANUM
0009            COMMON /COMU3/ DELTA
0010            PW(1,1) = 0.0
0011            PW(1,2) = 1.0
0012            PW(1,3) = 0.0
0013            PW(2,1) = 0.0
0014            PW(2,2) = 0.0
0015            PW(2,3) = 1.0
0016            IF(X .GT. TRPT) GO TO 25
0017            PW(3,1) = - Y(1,3)/( 1. + 2.*EPONU)
0018            PW(3,2) = - 2.*LAMBDA*Y(1,2)/(1. + 2.*EPONU)
0019              DENUM = -Y(1,1) - 2.*BETA*X*TERM1*TERM2*Y(1,3)
0020            ABA1 = 1.D00 + 2.D00*EPONU
0021            DDEN1 = 2.D00*EPONU/Y(1,3)
0022            PW(3,3) = (ABA1*DENUM - ANUM*DDEN1)/ABA1**2
               1 - ANUM*DDEN1/ABA1**2
0023            GO TO 30
0024      25    CONTINUE
0025            PW(3,1) = -Y(1,3)/(1. + EPONU)
0026            PW(3,2) = -2.*LAMBDA*Y(1,2)/(1. + EPONU)
0027            ABA2 = 1.D00 + EPONU
0028            DDEN2 = EPONU/Y(1,3)
0029             ARR2 = (-Y(1,1)*Y(1,3)
               1  + LAMBDA*((1.D00 - Y(1,2))**2))
0030            PW(3,3) =(ABA2*(-Y(1,1)) - ARR2*DDEN2)/ABA2**2
               1   - DELTA*DDEN2/ABA2**2
0031      30    CONTINUE
0032            RETURN
0033            END
```

COEFFICIENTS

0.0	0.1000D 01	0.0
-0.1006D 01	0.7755D 00	0.2730D-02
0.6809D 00	0.1295D-01	-0.1330D-03
0.1345D-01	-0.3200D-03	0.1163D-05
0.0	0.1000D 01	0.0
-0.1023D 01	0.7743D 00	0.2547D-02
0.6826D 00	0.1193D-01	-0.1130D-03
0.1228D-01	-0.2656D-03	0.8251D-06
0.0	0.1000D 01	0.0
-0.1169D 01	0.7800D 00	0.2238D-02
0.6831D 00	0.1124D-01	-0.1004D-03
0.1180D-01	-0.2486D-03	0.8559D-06

```
PROBLEM 5.3  THE TURBULENT BOUNDARY LAYER ON A FLAT PLATE


OUTPUT AT INTERVALS OF  0.50D 00
NUMBER OF EQUATIONS IS  3
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.5000000 00
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.3400000 02
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.0            0.0            0.5400D 01



FOR EPS =      0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.5155388D-02

THE ORDER OF THE INTEGRATION IS  6




FOR EPS =      0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.9849942D-02

THE ORDER OF THE INTEGRATION IS  4

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.7567075D-02


*****************************************
*METHOD SELECTED IS ADAM'S PREDICTOR CORRECTOR*
*****************************************
```

CE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2)	....
380	66	0.20313579D-01	0.50000000D 00	0.21921083D 00	0.55249978D 00	0.17849157D 00
460	83	0.23151799D-01	0.10000000D 01	0.51235205D 00	0.61281399D 00	0.85998567D-01
508	96	0.49188259D-01	0.15000000D 01	0.82805732D 00	0.64765273D 00	0.57250238D-01
536	104	0.95023046D-02	0.20000000D 01	0.11583534D 01	0.67234776D 00	0.42889804D-01
563	111	0.20542167D 00	0.25000000D 01	0.14994882D 01	0.69147529D 00	0.34259135D-01
598	118	0.24322292D 00	0.30000000D 01	0.18492465D 01	0.70707575D 00	0.28495175D-01
634	126	0.49480502D-01	0.35000000D 01	0.22061611D 01	0.72023974D 00	0.24371864D-01
654	132	0.85406835D-01	0.40000000D 01	0.25651899D 01	0.73161738D 00	0.21276188D-01
682	139	0.39324146D 00	0.45000000D 01	0.29375531D 01	0.74162849D 00	0.18867270D-01
706	145	0.10675854D 00	0.50000000D 01	0.33106411D 01	0.75056288D 00	0.16937765D-01
718	147	0.50000000D 00	0.55000000D 01	0.36879734D 01	0.75862288D 00	0.15359287D-01
738	152	0.12500000D 00	0.60000000D 01	0.40691474D 01	0.76596368D 00	0.14041873D-01
756	157	0.12500000D 00	0.65000000D 01	0.44538365D 01	0.77270352D 00	0.13026143D-01
776	162	0.12500000D 00	0.70000000D 01	0.48418149D 01	0.77920619D 00	0.12975754D-01
786	167	0.12500000D 00	0.75000000D 01	0.52330371D 01	0.78567719D 00	0.12908908D-01
796	172	0.12500000D 00	0.80000000D 01	0.56274867D 01	0.79211573D 00	0.12845833D-01
806	177	0.12500000D 00	0.85000000D 01	0.60251477D 01	0.79852358D 00	0.12786101D-01
816	182	0.12500000D 00	0.90000000D 01	0.64260054D 01	0.80490233D 00	0.12729354D-01
826	187	0.12500000D 00	0.95000000D 01	0.68300454D 01	0.81125338D 00	0.12675286D-01
836	192	0.12500000D 00	0.10000000D 02	0.72372544D 01	0.81757802D 00	0.12623638D-01
846	197	0.12500000D 00	0.10500000D 02	0.76476193D 01	0.82387738D 00	0.12574185D-01
856	202	0.12500000D 00	0.11000000D 02	0.80611277D 01	0.83015253D 00	0.12526733D-01
866	207	0.12500000D 00	0.11500000D 02	0.84777679D 01	0.83640442D 00	0.12481112D-01
876	212	0.12500000D 00	0.12000000D 02	0.88975284D 01	0.84263393D 00	0.12437174D-01
886	217	0.12500000D 00	0.12500000D 02	0.93203982D 01	0.84884185D 00	0.12394786D-01
896	222	0.12500000D 00	0.13000000D 02	0.97463668D 01	0.85502895D 00	0.12353833D-01
906	227	0.12500000D 00	0.13500000D 02	0.10175424D 02	0.86119591D 00	0.12314210D-01
916	232	0.12500000D 00	0.14000000D 02	0.10607559D 02	0.86734337D 00	0.12275824D-01
926	237	0.12500000D 00	0.14500000D 02	0.11042764D 02	0.87347192D 00	0.12238592D-01
936	242	0.12500000D 00	0.15000000D 02	0.11481028D 02	0.87958214D 00	0.12202439D-01
946	247	0.12500000D 00	0.15500000D 02	0.11922343D 02	0.88567453D 00	0.12167295D-01
956	252	0.12500000D 00	0.16000000D 02	0.12366700D 02	0.89174959D 00	0.12133099D-01
966	257	0.12500000D 00	0.16500000D 02	0.12814090D 02	0.89780778D 00	0.12099796D-01
976	262	0.12500000D 00	0.17000000D 02	0.13264505D 02	0.90384953D 00	0.12067332D-01
986	267	0.12500000D 00	0.17500000D 02	0.13717937D 02	0.90987524D 00	0.12035662D-01
996	272	0.12500000D 00	0.18000000D 02	0.14174378D 02	0.91588531D 00	0.12004742D-01
1006	277	0.12500000D 00	0.18500000D 02	0.14633820D 02	0.92188010D 00	0.11974531D-01
1016	282	0.12500000D 00	0.19000000D 02	0.15096255D 02	0.92785996D 00	0.11944995D-01
1026	287	0.12500000D 00	0.19500000D 02	0.15561677D 02	0.93382520D 00	0.11916097D-01
1036	292	0.12500000D 00	0.20000000D 02	0.16030078D 02	0.93977616D 00	0.11887808D-01
1046	297	0.12500000D 00	0.20500000D 02	0.16501451D 02	0.94571311D 00	0.11860098D-01
1056	302	0.12500000D 00	0.21000000D 02	0.16975789D 02	0.95163635D 00	0.11832939D-01
1066	307	0.12500000D 00	0.21500000D 02	0.17453085D 02	0.95754614D 00	0.11806307D-01
1076	312	0.12500000D 00	0.22000000D 02	0.17933333D 02	0.96344274D 00	0.11780178D-01
1086	317	0.12500000D 00	0.22500000D 02	0.18416526D 02	0.96932639D 00	0.11754530D-01
1096	322	0.12500000D 00	0.23000000D 02	0.18902657D 02	0.97519734D 00	0.11729343D-01
1106	327	0.12500000D 00	0.23500000D 02	0.19391721D 02	0.98105581D 00	0.11704597D-01
1116	332	0.12500000D 00	0.24000000D 02	0.19883711D 02	0.98690201D 00	0.11680275D-01
1126	337	0.12500000D 00	0.24500000D 02	0.20378621D 02	0.99273615D 00	0.11656358D-01
1136	342	0.12500000D 00	0.25000000D 02	0.20876445D 02	0.99855843D 00	0.11632832D-01
1146	347	0.12500000D 00	0.25500000D 02	0.21377177D 02	0.10043690D 01	0.11609681D-01
1156	352	0.12500000D 00	0.26000000D 02	0.21880812D 02	0.10101682D 01	0.11586891D-01
1166	357	0.12500000D 00	0.26500000D 02	0.22387344D 02	0.10159560D 01	0.11564449D-01

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2)	
1176	362	0.12500000D 00	0.27000000D 02	0.22856766D 02	0.10217327D 01	0.11542342D-01
1186	367	0.12500000D 00	0.27500000D 02	0.23409075D 02	0.10274984D 01	0.11520558D-01
1196	372	0.12500000D 00	0.28000000D 02	0.23924263D 02	0.10332533D 01	0.11499086D-01
1206	377	0.12500000D 00	0.28500000D 02	0.24442326D 02	0.10389975D 01	0.11477914D-01
1216	382	0.12500000D 00	0.29000000D 02	0.24963259D 02	0.10447313D 01	0.11457034D-01
1226	387	0.12500000D 00	0.29500000D 02	0.25487056D 02	0.10504546D 01	0.11436434D-01
1236	392	0.12500000D 00	0.30000000D 02	0.26013712D 02	0.10561677D 01	0.11416106D-01
1246	397	0.12500000D 00	0.30500000D 02	0.26543222D 02	0.10618708D 01	0.11396042D-01
1256	402	0.12500000D 00	0.31000000D 02	0.27075581D 02	0.10675638D 01	0.11376232D-01
1266	407	0.12500000D 00	0.31500000D 02	0.27610784D 02	0.10732470D 01	0.11356669D-01
1276	412	0.12500000D 00	0.32000000D 02	0.28148826D 02	0.10789205D 01	0.11337346D-01
1286	417	0.12500000D 00	0.32500000D 02	0.28689703D 02	0.10845844D 01	0.11318255D-01
1296	422	0.12500000D 00	0.33000000D 02	0.29233409D 02	0.10902388D 01	0.11299388D-01
1306	427	0.12500000D 00	0.33500000D 02	0.29779940D 02	0.10958838D 01	0.11280741D-01
1316	432	0.12500000D 00	0.34000000D 02	0.30329291D 02	0.11015196D 01	0.11262305D-01

TOTAL NUMBER OF FUNCTION EVALUATIONS 1318

```
      SUBROUTINE DIFFUN(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,2),DY(2)
      DATA BPERID / 6.D00/
      DATA DOSPI / 6.28318D00/
      COMMON NCFNS
      NCFNS = NCFNS + 1
      ARGUM = DOSPI*X/BPERID
      AF = DCOS(ARGUM)
      TC = 390.D00 + 10.D00*AF
      DY(1) = 6.5D00 - Y(1,1) - 1.75754943D14*
     |  DEXP(-14088.D00/Y(1,2))*Y(1,1)
      DY(2) = 2.162D00*(TC - Y(1,2))  +
     |   4.745383461D15*DEXP(- 14088.D00/Y(1,2))*Y(1,1)
      RETURN
      END
```

```
      SUBROUTINE PEDERV( X,Y,PW,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,2),PW(2,2)
      PW(1,1) = - 1.D00 - 1.75754943D14*DEXP( - 14088.D00/Y(1,2))
      PW(1,2) = - 1.75754943D14*DEXP(-14088.D00/Y(1,2))*Y(1,1)*
     1 (14088.D00/(Y(1,2)**2))
      PW(2,1) = 4.74538346D15*DEXP(-14088.D00/Y(1,2))
      PW(2,2) = - 2.162D00          + 4.745383461D15*
     1 DEXP( - 14088.D00/Y(1,2))*Y(1,1)*
     1 (14088.D00/(Y(1,2)**2))
      RETURN
      END
```

```
PROBLEM 3.4.A THE PERIODIC CHEMICAL REACTION (SINUSOIDAL FORCING FUNCTION


OUTPUT AT INTERVALS OF  0.25D 00
NUMBER OF EQUATIONS IS  2
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.250000D 00
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.180000D 02
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.1929D 01    0.4265D 03



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.2630809D-01

THE ORDER OF THE INTEGRATION IS  5




FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.6279452D-01

THE ORDER OF THE INTEGRATION IS  3

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.3067298D-01


**************************************************
*METHOD SELECTED IS ADAM'S PREDICTOR CORRECTOR*
**************************************************
```

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
24	10	0.67300585D-03	0.25000000D 00	0.25881950D 01	0.42366863D 03
40	16	0.19694432D-01	0.50000000D 00	0.30755424D 01	0.42236204D 03
52	19	0.89726630D-01	0.75000000D 00	0.34311116D 01	0.42160943D 03
61	22	0.12055662D 00	0.10000000D 01	0.36977255D 01	0.42062426D 03
70	25	0.12055662D 00	0.12500000D 01	0.39254698D 01	0.41860691D 03
85	29	0.62418482D-01	0.15000000D 01	0.41640146D 01	0.41491334D 03
96	33	0.62418482D-01	0.17500000D 01	0.44383524D 01	0.40958030D 03
108	37	0.62418482D-01	0.20000000D 01	0.47356203D 01	0.40346740D 03
119	41	0.62418482D-01	0.22500000D 01	0.50258897D 01	0.39760244D 03
130	45	0.62418482D-01	0.25000000D 01	0.52870216D 01	0.39261144D 03
140	49	0.62418482D-01	0.27500000D 01	0.55100671D 01	0.38871480D 03
148	53	0.62418482D-01	0.30000000D 01	0.56946249D 01	0.38592413D 03
155	57	0.62418482D-01	0.32500000D 01	0.58441373D 01	0.38417738D 03
163	61	0.62418482D-01	0.35000000D 01	0.59631958D 01	0.38339332D 03
171	65	0.62418482D-01	0.37500000D 01	0.60562581D 01	0.38348428D 03
179	69	0.62418482D-01	0.40000000D 01	0.61270891D 01	0.38435472D 03
187	73	0.62418482D-01	0.42500000D 01	0.61785295D 01	0.38589836D 03
195	77	0.62418482D-01	0.45000000D 01	0.62124026D 01	0.38759805D 03
203	81	0.62418482D-01	0.47500000D 01	0.62294802D 01	0.39052918D 03
211	85	0.62418482D-01	0.50000000D 01	0.62294988D 01	0.39336562D 03
219	89	0.62418482D-01	0.52500000D 01	0.62112511D 01	0.39638621D 03
226	93	0.62418482D-01	0.55000000D 01	0.61728074D 01	0.39947997D 03
233	97	0.62418482D-01	0.57500000D 01	0.61119288D 01	0.40254761D 03
241	101	0.62418482D-01	0.60000000D 01	0.60267417D 01	0.40549760D 03
249	105	0.62418482D-01	0.62500000D 01	0.59167814D 01	0.40823405D 03
257	109	0.62418482D-01	0.65000000D 01	0.57845894D 01	0.41063217D 03
267	113	0.62418482D-01	0.67500000D 01	0.56381745D 01	0.41249496D 03
277	117	0.62418482D-01	0.70000000D 01	0.54943630D 01	0.41349297D 03
288	121	0.62418482D-01	0.72500000D 01	0.53809350D 01	0.41314005D 03
300	125	0.62418482D-01	0.75000000D 01	0.53303308D 01	0.41096913D 03
311	129	0.62418482D-01	0.77500000D 01	0.53595543D 01	0.40690870D 03
321	133	0.62418482D-01	0.80000000D 01	0.54547930D 01	0.40192074D 03
332	137	0.62418482D-01	0.82500000D 01	0.55840691D 01	0.39674746D 03
343	141	0.62418482D-01	0.85000000D 01	0.57194713D 01	0.39216657D 03
353	145	0.62418482D-01	0.87500000D 01	0.58450691D 01	0.38849760D 03
361	149	0.62418482D-01	0.90000000D 01	0.59542171D 01	0.38582822D 03
369	153	0.62418482D-01	0.92500000D 01	0.60493457D 01	0.38414406D 03
376	157	0.62418482D-01	0.95000000D 01	0.61191583D 01	0.38339124D 03
382	161	0.62418482D-01	0.97500000D 01	0.61771213D 01	0.38349714D 03
390	165	0.62418482D-01	0.10000000D 02	0.62206983D 01	0.38437440D 03
398	169	0.62418482D-01	0.10250000D 02	0.62509559D 01	0.38592104D 03
406	173	0.62418482D-01	0.10500000D 02	0.62683456D 01	0.38802215D 03
414	177	0.62418482D-01	0.10750000D 02	0.62725769D 01	0.39055433D 03
422	181	0.62418482D-01	0.11000000D 02	0.62625606D 01	0.39339210D 03
429	185	0.62418482D-01	0.11250000D 02	0.62364454D 01	0.39641473D 03
435	189	0.62418482D-01	0.11500000D 02	0.61917955D 01	0.39951156D 03
443	193	0.62418482D-01	0.11750000D 02	0.61259679D 01	0.40258373D 03
451	197	0.62418482D-01	0.12000000D 02	0.60367584D 01	0.40554037D 03
459	201	0.62418482D-01	0.12250000D 02	0.59234152D 01	0.40828665D 03
467	205	0.62418482D-01	0.12500000D 02	0.57882131D 01	0.41069944D 03
477	209	0.62418482D-01	0.12750000D 02	0.56389005D 01	0.41258394D 03
488	213	0.62418482D-01	0.13000000D 02	0.54920928D 01	0.41361212D 03
499	217	0.62418482D-01	0.13250000D 02	0.53755911D 01	0.41329411D 03

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
511	221	0.62418482D-01	0.13500000D 02	0.53223563D 01	0.41114867D 03
522	225	0.62418482D-01	0.13750000D 02	0.53501950D 01	0.40716692D 03
532	229	0.62418482D-01	0.14000000D 02	0.54455211D 01	0.40207065D 03
543	233	0.62418482D-01	0.14250000D 02	0.55758264D 01	0.39685845D 03
554	237	0.62418482D-01	0.14500000D 02	0.57125747D 01	0.39224227D 03
564	241	0.62418482D-01	0.14750000D 02	0.58394797D 01	0.38854675D 03
572	245	0.62418482D-01	0.15000000D 02	0.59497620D 01	0.38585922D 03
580	249	0.62418482D-01	0.15250000D 02	0.60418265D 01	0.38416326D 03
587	253	0.62418482D-01	0.15500000D 02	0.61163927D 01	0.38340299D 03
593	257	0.62418482D-01	0.15750000D 02	0.61749548D 01	0.38350427D 03
601	261	0.62418482D-01	0.16000000D 02	0.62190050D 01	0.38437867D 03
609	265	0.62418482D-01	0.16250000D 02	0.62496348D 01	0.38592355D 03
617	269	0.62418482D-01	0.16500000D 02	0.62673170D 01	0.38802358D 03
625	273	0.62418482D-01	0.16750000D 02	0.62717780D 01	0.39055509D 03
633	277	0.62418482D-01	0.17000000D 02	0.62619423D 01	0.39339243D 03
640	281	0.62418482D-01	0.17250000D 02	0.62359695D 01	0.39641476D 03
646	285	0.62418482D-01	0.17500000D 02	0.61914324D 01	0.39951139D 03
654	289	0.62418482D-01	0.17750000D 02	0.61256951D 01	0.40258338D 03
662	293	0.62418482D-01	0.18000000D 02	0.60365590D 01	0.40553985D 03

TOTAL NUMBER OF FUNCTION EVALUATIONS 664

```
0001            SUBROUTINE DIFFUN(X,Y,DY)
0002            IMPLICIT REAL*8 (A-H,O-Z)
0003            DIMENSION Y(8,2),DY(2)
0004            DATA BPERID / 6.0D0/
0005            DATA DOSPI / 6.28318E00/
0006            COMMON NOFNS
0007            NOFNS = NOFNS + 1
0008            TO = 380.0D0
0009            ARGUM = DOSPI*X/BPERID
0010            AF = DCOS(ARGUM)
0011            IF ( AF .LT. 0.0D0) TO = 400.0D0
0012            DY(1) = 6.5D00 - Y(1,1) - 1.75754943D14*
               1  DEXP(-14088.D00/Y(1,2))*Y(1,1)
0013            DY(2) = 2.162D00*(TO - Y(1,2))  +
               1    4.745383461D15*DEXP(- 14088.D00/Y(1,2))*Y(1,1)
0014            RETURN
0015            END
```

```
0001      SUBROUTINE PEDERV( X,Y,PW,M)
0002      IMPLICIT REAL*8 (A-H,O-Z)
0003      DIMENSION Y(8,2),PW(2,2)
0004      PW(1,1) = - 1.DOO - 1.75754943D14*DEXP( - 14088.DOO/Y(1,2))
0005      PW(1,2) = - 1.75754943D14*DEXP(-14088.DOO/Y(1,2))*Y(1,1)*
         1 (14088.DOO/(Y(1,2)**2))
0006      PW(2,1) = 4.74538346D15*DEXP(-14088.DOO/Y(1,2))
0007      PW(2,2) = - 2.162DOO         + 4.745383461D15*
         1 DEXP( - 14088.DOO/Y(1,2))*Y(1,1)*
         1 (14088.DOO/(Y(1,2)**2))
0008      RETURN
0009      END
```

PROBLEM 3.4.8 THE PERIODIC CHEMICAL REACTION (BANG-BANG FORCING FUNCTION)

```
OUTPUT AT INTERVALS OF  0.10D 00
NUMBER OF EQUATIONS IS  2
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-03
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000D 00
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.1800000D 02
EXACT INTERVALS WILL BE PROVIDED
THE USER HAS SELECTED THE INTEGRATION METHOD

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.1929D 01    0.4265D 03


*********************************
*METHOD FLAG SELECTED BY USER IS   0*
*********************************
```

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
11	6	0.22576660D-01	0.1000000D 00	0.22388410D 01	0.42061088D 03
19	10	0.22776660D-01	0.2000000D 00	0.25533706D 01	0.41502833D 03
27	14	0.22776660D-01	0.3000000D 00	0.28626161D 01	0.40990430D 03
35	18	0.22776660D-01	0.4000000D 00	0.31596518D 01	0.40533878D 03
43	22	0.22776660D-01	0.5000000D 00	0.34402285D 01	0.40136084D 03
50	26	0.22776660D-01	0.6000000D 00	0.37021867D 01	0.39795095D 03
56	30	0.22776660D-01	0.7000000D 00	0.39447942D 01	0.39506221D 03
63	34	0.22776660D-01	0.8000000D 00	0.41682218D 01	0.39263597D 03
70	38	0.22776660D-01	0.9000000D 00	0.43731653D 01	0.39061091D 03
77	42	0.22776660D-01	0.1000000D 01	0.45606128D 01	0.38892862D 03
84	46	0.22776660D-01	0.1100000D 01	0.47316926D 01	0.38753608D 03
91	50	0.22776660D-01	0.1200000D 01	0.48875820D 01	0.38638661D 03
98	54	0.22776660D-01	0.1300000D 01	0.50294522D 01	0.38543993D 03
105	58	0.22776660D-01	0.1400000D 01	0.51584359D 01	0.38466171D 03
136	67	0.63312096D-03	0.1500000D 01	0.52755475D 01	0.38402369D 03
148	73	0.55100450D-01	0.1600000D 01	0.53791119D 01	0.38742597D 03
154	75	0.10000000D 00	0.1700000D 01	0.54689380D 01	0.39031230D 03
160	77	0.10000000D 00	0.1800000D 01	0.55452006D 01	0.39276371D 03
166	79	0.10000000D 00	0.1900000D 01	0.56088258D 01	0.39487431D 03
172	81	0.10000000D 00	0.2000000D 01	0.56607519D 01	0.39671737D 03
178	83	0.10000000D 00	0.2100000D 01	0.57019003D 01	0.39835004D 03
184	85	0.10000000D 00	0.2200000D 01	0.57331508D 01	0.39981711D 03
190	87	0.10000000D 00	0.2300000D 01	0.57553219D 01	0.40115409D 03
196	89	0.10000000D 00	0.2400000D 01	0.57691577D 01	0.40238944D 03
202	91	0.10000000D 00	0.2500000D 01	0.57753176D 01	0.40354642D 03
208	93	0.10000000D 00	0.2600000D 01	0.57743694D 01	0.40464451D 03
214	95	0.10000000D 00	0.2700000D 01	0.57667831D 01	0.40570050D 03
220	97	0.10000000D 00	0.2800000D 01	0.57529239D 01	0.40672942D 03
226	99	0.10000000D 00	0.2900000D 01	0.57330430D 01	0.40774543D 03
232	101	0.10000000D 00	0.3000000D 01	0.57072636D 01	0.40876255D 03
238	103	0.10000000D 00	0.3100000D 01	0.56755602D 01	0.40979559D 03
244	105	0.10000000D 00	0.3200000D 01	0.56377248D 01	0.41086121D 03
250	107	0.10000000D 00	0.3300000D 01	0.55933131D 01	0.41197936D 03
256	109	0.10000000D 00	0.3400000D 01	0.55415572D 01	0.41317547D 03
262	111	0.10000000D 00	0.3500000D 01	0.54812162D 01	0.41448395D 03
268	113	0.10000000D 00	0.3600000D 01	0.54103102D 01	0.41595447D 03
282	118	0.25000000D-01	0.3700000D 01	0.53260692D 01	0.41765337D 03
286	120	0.10000000D 00	0.3800000D 01	0.52229317D 01	0.41971045D 03
298	125	0.25000000D-01	0.3900000D 01	0.50914078D 01	0.42234635D 03
308	130	0.25000000D-01	0.4000000D 01	0.49111650D 01	0.42604026D 03
322	135	0.25000000D-01	0.4100000D 01	0.46282824D 01	0.43209588D 03
357	152	0.32840126D-02	0.4200000D 01	0.39913017D 01	0.44683617D 03
657	277	0.22945247D-02	0.4300000D 01	0.61266104D-02	0.54522647D 03
920	359	0.13563577D-02	0.4400000D 01	0.11207754D-01	0.53265219D 03
1078	412	0.11891646D-02	0.4500000D 01	0.18699089D-01	0.52244847D 03
1175	447	0.31504844D-02	0.4600000D 01	0.35173621D-01	0.51009242D 03
1216	464	0.63946695D-02	0.4700000D 01	0.60723085D-01	0.49984117D 03
1233	473	0.13574406D-01	0.4800000D 01	0.96972327D-01	0.49124705D 03
1244	479	0.23581077D-01	0.4900000D 01	0.14527743D 00	0.48392716D 03
1256	485	0.18650949D-02	0.5000000D 01	0.20665986D 00	0.47757886D 03
1267	491	0.55952847D-02	0.5100000D 01	0.28210870D 00	0.47195660D 03
1277	496	0.25000000D-01	0.5200000D 01	0.37277394D 00	0.46685545D 03
1287	501	0.25000000D-01	0.5300000D 01	0.48030772D 00	0.46209482D 03

CE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
1297	506	0.25000000D-01	0.5400000D 01	0.60713703D 00	0.45750557D 03
1303	508	0.10000000D 00	0.5500000D 01	0.75652329D 00	0.45292581D 03
1309	510	0.10000000D 00	0.5600000D 01	0.93258140D 00	0.44819044D 03
1315	512	0.10000000D 00	0.5700000D 01	0.11392436D 01	0.44316566D 03
1321	514	0.10000000D 00	0.5800000D 01	0.13787280D 01	0.43777722D 03
1327	516	0.10000000D 00	0.5900000D 01	0.16495255D 01	0.43205565D 03
1333	518	0.10000000D 00	0.6000000D 01	0.19453397D 01	0.42615172D 03
1337	520	0.10000000D 00	0.6100000D 01	0.22561183D 01	0.42029777D 03
1343	522	0.10000000D 00	0.6200000D 01	0.25707524D 01	0.41473167D 03
1349	524	0.10000000D 00	0.6300000D 01	0.28796644D 01	0.40963235D 03
1355	526	0.10000000D 00	0.6400000D 01	0.31760186D 01	0.40509671D 03
1361	528	0.10000000D 00	0.6500000D 01	0.34556846D 01	0.40115045D 03
1367	530	0.10000000D 00	0.6600000D 01	0.37166084D 01	0.39777144D 03
1373	532	0.10000000D 00	0.6700000D 01	0.39581374D 01	0.39491122D 03
1377	534	0.10000000D 00	0.6800000D 01	0.41804908D 01	0.39251024D 03
1381	536	0.10000000D 00	0.6900000D 01	0.43843971D 01	0.39050702D 03
1385	538	0.10000000D 00	0.7000000D 01	0.45708627D 01	0.38884324D 03
1389	540	0.10000000D 00	0.7100000D 01	0.47410247D 01	0.38746621D 03
1393	542	0.10000000D 00	0.7200000D 01	0.48960637D 01	0.38632963D 03
1397	544	0.10000000D 00	0.7300000D 01	0.50371505D 01	0.38539359D 03
1401	546	0.10000000D 00	0.7400000D 01	0.51654158D 01	0.38462412D 03
1405	548	0.10000000D 00	0.7500000D 01	0.52819326D 01	0.38399258D 03
1425	555	0.14110744D-02	0.7600000D 01	0.53860172D 01	0.38693648D 03
1431	557	0.10000000D 00	0.7700000D 01	0.54758556D 01	0.38989747D 03
1437	559	0.10000000D 00	0.7800000D 01	0.55521224D 01	0.39240264D 03
1443	561	0.10000000D 00	0.7900000D 01	0.56158048D 01	0.39455703D 03
1447	563	0.10000000D 00	0.8000000D 01	0.56678242D 01	0.39643574D 03
1451	565	0.10000000D 00	0.8100000D 01	0.57090914D 01	0.39809745D 03
1455	567	0.10000000D 00	0.8200000D 01	0.57404760D 01	0.39958811D 03
1459	569	0.10000000D 00	0.8300000D 01	0.57627932D 01	0.40094419D 03
1463	571	0.10000000D 00	0.8400000D 01	0.57767859D 01	0.40219489D 03
1467	573	0.10000000D 00	0.8500000D 01	0.57831151D 01	0.40336405D 03
1471	575	0.10000000D 00	0.8600000D 01	0.57823524D 01	0.40447155D 03
1475	577	0.10000000D 00	0.8700000D 01	0.57749733D 01	0.40553447D 03
1479	579	0.10000000D 00	0.8800000D 01	0.57613512D 01	0.40656803D 03
1483	581	0.10000000D 00	0.8900000D 01	0.57417480D 01	0.40758642D 03
1487	583	0.10000000D 00	0.9000000D 01	0.57163021D 01	0.40860362D 03
1491	585	0.10000000D 00	0.9100000D 01	0.56850079D 01	0.40963422D 03
1495	587	0.10000000D 00	0.9200000D 01	0.56476857D 01	0.41069443D 03
1499	589	0.10000000D 00	0.9300000D 01	0.56039329D 01	0.41180348D 03
1503	591	0.10000000D 00	0.9400000D 01	0.55530444D 01	0.41298557D 03
1507	593	0.10000000D 00	0.9500000D 01	0.54938802D 01	0.41427304D 03
1513	595	0.10000000D 00	0.9600000D 01	0.54246312D 01	0.41571187D 03
1519	597	0.10000000D 00	0.9700000D 01	0.53423829D 01	0.41737182D 03
1525	599	0.10000000D 00	0.9800000D 01	0.52422260D 01	0.41936736D 03
1537	604	0.25000000D-01	0.9900000D 01	0.51156428D 01	0.42189632D 03
1547	609	0.25000000D-01	0.1000000D 02	0.49445535D 01	0.42538363D 03
1560	614	0.25000000D-01	0.1010000D 02	0.46833956D 01	0.43092586D 03
1599	629	0.45313582D-02	0.1020000D 02	0.41460796D 01	0.44316866D 03
1863	742	0.10948003D-02	0.1030000D 02	0.55106474D-02	0.54737436D 03
2132	834	0.20915443D-02	0.1040000D 02	0.10276701D-01	0.53439034D 03
2311	896	0.17681087D-02	0.1050000D 02	0.17401274D-01	0.52385876D 03
2426	937	0.49163822D-02	0.1060000D 02	0.33114453D-01	0.51123618D 03

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
2468	955	0.89087070D-02	0.10700000D 02	0.57678155D-01	0.50079277D 03
2486	964	0.35695793D-02	0.10800000D 02	0.92785015D-01	0.49205020D 03
2498	971	0.12236429D-01	0.10900000D 02	0.13982670D 00	0.48461694D 03
2508	977	0.19893379D-01	0.11000000D 02	0.19985068D 00	0.47818272D 03
2519	982	0.25000000D-01	0.11100000D 02	0.27382755D 00	0.47249706D 03
2530	987	0.25000000D-01	0.11200000D 02	0.36288936D 00	0.46735188D 03
2541	992	0.25000000D-01	0.11300000D 02	0.46862824D 00	0.46256461D 03
2553	997	0.25000000D-01	0.11400000D 02	0.59338242D 00	0.45796528D 03
2564	1002	0.25000000D-01	0.11500000D 02	0.74038505D 00	0.45338848D 03
2570	1004	0.10000000D 00	0.11600000D 02	0.91355539D 00	0.44867694D 03
2576	1006	0.10000000D 00	0.11700000D 02	0.11170113D 01	0.44368572D 03
2582	1008	0.10000000D 00	0.11800000D 02	0.13532043D 01	0.43833422D 03
2588	1010	0.10000000D 00	0.11900000D 02	0.16210735D 01	0.43264102D 03
2594	1012	0.10000000D 00	0.12000000D 02	0.19147935D 01	0.42674563D 03
2598	1014	0.10000000D 00	0.12100000D 02	0.22245821D 01	0.42087537D 03
2604	1016	0.10000000D 00	0.12200000D 02	0.25393042D 01	0.41527102D 03
2610	1018	0.10000000D 00	0.12300000D 02	0.28491492D 01	0.41011917D 03
2616	1020	0.10000000D 00	0.12400000D 02	0.31469916D 01	0.40552484D 03
2622	1022	0.10000000D 00	0.12500000D 02	0.34284539D 01	0.40151988D 03
2628	1024	0.10000000D 00	0.12600000D 02	0.36913063D 01	0.39808588D 03
2634	1026	0.10000000D 00	0.12700000D 02	0.39347825D 01	0.39517625D 03
2638	1028	0.10000000D 00	0.12800000D 02	0.41590330D 01	0.39273201D 03
2642	1030	0.10000000D 00	0.12900000D 02	0.43647478D 01	0.39069162D 03
2646	1032	0.10000000D 00	0.13000000D 02	0.45529131D 01	0.38899629D 03
2650	1034	0.10000000D 00	0.13100000D 02	0.47246576D 01	0.38759271D 03
2654	1036	0.10000000D 00	0.13200000D 02	0.48811603D 01	0.38643393D 03
2658	1038	0.10000000D 00	0.13300000D 02	0.50235949D 01	0.38547940D 03
2662	1040	0.10000000D 00	0.13400000D 02	0.51530969D 01	0.38469461D 03
2666	1042	0.10000000D 00	0.13500000D 02	0.52707456D 01	0.38405039D 03
2686	1049	0.14097333D-02	0.13600000D 02	0.53758599D 01	0.38698395D 03
2692	1051	0.10000000D 00	0.13700000D 02	0.54666290D 01	0.38993664D 03
2698	1053	0.10000000D 00	0.13800000D 02	0.55437382D 01	0.39243509D 03
2704	1055	0.10000000D 00	0.13900000D 02	0.56081851D 01	0.39458403D 03
2708	1057	0.10000000D 00	0.14000000D 02	0.56608997D 01	0.39645825D 03
2712	1059	0.10000000D 00	0.14100000D 02	0.57028003D 01	0.39811623D 03
2716	1061	0.10000000D 00	0.14200000D 02	0.57347630D 01	0.39960377D 03
2720	1063	0.10000000D 00	0.14300000D 02	0.57576084D 01	0.40095720D 03
2724	1065	0.10000000D 00	0.14400000D 02	0.57720846D 01	0.40220562D 03
2728	1067	0.10000000D 00	0.14500000D 02	0.57788570D 01	0.40337280D 03
2732	1069	0.10000000D 00	0.14600000D 02	0.57785011D 01	0.40447856D 03
2736	1071	0.10000000D 00	0.14700000D 02	0.57714962D 01	0.40553992D 03
2740	1073	0.10000000D 00	0.14800000D 02	0.57582189D 01	0.40657206D 03
2744	1075	0.10000000D 00	0.14900000D 02	0.57389346D 01	0.40758914D 03
2748	1077	0.10000000D 00	0.15000000D 02	0.57137847D 01	0.40860509D 03
2752	1079	0.10000000D 00	0.15100000D 02	0.56827667D 01	0.40963446D 03
2756	1081	0.10000000D 00	0.15200000D 02	0.56457043D 01	0.41069344D 03
2760	1083	0.10000000D 00	0.15300000D 02	0.56021984D 01	0.41180119D 03
2764	1085	0.10000000D 00	0.15400000D 02	0.55515483D 01	0.41298184D 03
2768	1087	0.10000000D 00	0.15500000D 02	0.54926196D 01	0.41426766D 03
2774	1089	0.10000000D 00	0.15600000D 02	0.54236114D 01	0.41570444D 03
2780	1091	0.10000000D 00	0.15700000D 02	0.53416220D 01	0.41736165D 03
2786	1093	0.10000000D 00	0.15800000D 02	0.52417656D 01	0.41935336D 03
2798	1098	0.25000000D-01	0.15900000D 02	0.51155704D 01	0.42187623D 03

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
2808	1103	0.25000000D-01	0.1600000D 02	0.49450772D 01	0.42535232D 03
2821	1109	0.25000000D-01	0.1610000D 02	0.46851385D 01	0.43086797D 03
2840	1123	0.42205600D-02	0.1620000D 02	0.41525746D 01	0.44299445D 03
3126	1232	0.26159462D-03	0.1629848D 02	0.54406719D-02	0.54769288D 03
3417	1329	0.16100822D-02	0.1639962D 02	0.10215596D-01	0.53451634D 03
3567	1380	0.46804298D-02	0.1650000D 02	0.17363861D-01	0.52393017D 03
3670	1414	0.65554009D-02	0.1660000D 02	0.33006065D-01	0.51129961D 03
3712	1432	0.33717060D-02	0.1670000D 02	0.57526290D-01	0.50084523D 03
3740	1443	0.97038962D-02	0.1680000D 02	0.92561974D-01	0.49209486D 03
3751	1449	0.48457292D-02	0.1690000D 02	0.13952757D 00	0.48465481D 03
3762	1454	0.25000000D-01	0.1700000D 02	0.19948864D 00	0.47821526D 03
3773	1459	0.25000000D-01	0.1710000D 02	0.27338621D 00	0.47252620D 03
3784	1464	0.25000000D-01	0.1720000D 02	0.36236450D 00	0.46737856D 03
3795	1469	0.25000000D-01	0.1730000D 02	0.46808844D 00	0.46258980D 03
3807	1474	0.25000000D-01	0.1740000D 02	0.59265243D 00	0.45798990D 03
3818	1479	0.25000000D-01	0.1750000D 02	0.73952482D 00	0.45341341D 03
3824	1481	0.10000000D 00	0.1760000D 02	0.91254231D 00	0.44870303D 03
3830	1483	0.10000000D 00	0.1770000D 02	0.11158259D 01	0.44371359D 03
3836	1485	0.10000000D 00	0.1780000D 02	0.13518410D 01	0.43836410D 03
3842	1487	0.10000000D 00	0.1790000D 02	0.16195505D 01	0.43267247D 03
3848	1489	0.10000000D 00	0.1800000D 02	0.19131546D 01	0.42677762D 03
3852	1491	0.10000000D 00	0.1810000D 02	0.22228864D 01	0.42090655D 03

TOTAL NUMBER OF FUNCTION EVALUATIONS 3852

```
      SUBROUTINE DIFFUN(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,2),DY(2)
      DATA DK/3.D00/
C EPSLON IS EQUAL TO 1.0D00/98.D00
      DATA EPSLON / 0.010204081632D00/
      COMMON NOFNS
      NOFNS = NOFNS + 1
      DY(1) = -Y(1,1) - Y(1,1)*Y(1,2) + EPSLON*DK*Y(1,2)
      DY(2) =(Y(1,1) - Y(1,1)*Y(1,2) - EPSLON*DK*Y(1,2))/EPSLON
      RETURN
      END
```

```
      SUBROUTINE PEDERV( X,Y,PW,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,2),PW(2,2)
      DATA DK/3.D0/
C EPSLON IS EQUAL TO 1.0D00/98.D00
      DATA EPSLON / 0.0102040816320D0/
      PW(1,1) = - 1.D00 - Y(1,2)
      PW(1,2) = - Y(1,1) + EPSLON*DK
      PW(2,1) = ( 1.D00 - Y(1,2))/EPSLON
      PW(2,2) = - (Y(1,1) + EPSLON*DK)/EPSLON
      RETURN
      END
```

```
PROBLEM 3.5 THE THERMAL DECOMPOSITION OF OZONE


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  2
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000D 01
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.1000000D 03
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.10000D 01     0.0



FOR EPS =      0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.4203509D-02

THE ORDER OF THE INTEGRATION IS  4




FOR EPS =      0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.9119119D-02

THE ORDER OF THE INTEGRATION IS  3

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.1097182D-01


****************************************
METHOD SELECTED IS GEAR'S STIFF OPTION   1*
****************************************
```

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
266	106	0.25966726D-01	0.10000000D 01	0.15990569D 00	0.85020257D 00
305	119	0.39307504D-01	0.20000000D 01	0.38701923D-01	0.59694343D 00
331	129	0.12062752D 00	0.30000000D 01	0.16204240D-01	0.38166053D 00
354	137	0.15304123D 00	0.40000000D 01	0.95463053D-02	0.26011142D 00
370	143	0.11947065D 00	0.50000000D 01	0.66564850D-02	0.19262848D 00
386	149	0.11947065D 00	0.60000000D 01	0.50823259D-02	0.15172434D 00
398	155	0.11947065D 00	0.70000000D 01	0.41019794D-02	0.12472977D 00
409	161	0.11947065D 00	0.80000000D 01	0.34339446D-02	0.10571180D 00
418	167	0.11947065D 00	0.90000000D 01	0.29508519D-02	0.91638801D-01
424	173	0.11947065D 00	0.10000000D 02	0.25857070D-02	0.80825064D-01
430	179	0.11947065D 00	0.11000000D 02	0.23002484D-02	0.72266019D-01
436	185	0.11947065D 00	0.12000000D 02	0.20710859D-02	0.65328592D-01
442	191	0.11947065D 00	0.13000000D 02	0.18831369D-02	0.59594917D-01
448	197	0.11947065D 00	0.14000000D 02	0.17262475D-02	0.54778592D-01
454	203	0.11947065D 00	0.15000000D 02	0.15933363D-02	0.50676961D-01
460	209	0.11947065D 00	0.16000000D 02	0.14793165D-02	0.47142722D-01
466	215	0.11947065D 00	0.17000000D 02	0.13804415D-02	0.44066289D-01
472	221	0.11947065D 00	0.18000000D 02	0.12938916D-02	0.41364496D-01
478	227	0.11947065D 00	0.19000000D 02	0.12175045D-02	0.38973099D-01
484	233	0.11947065D 00	0.20000000D 02	0.11495946D-02	0.36841712D-01
490	239	0.11947065D 00	0.21000000D 02	0.10882889D-02	0.34930259D-01
496	245	0.11947065D 00	0.22000000D 02	0.10341392D-02	0.33206480D-01
502	251	0.11947065D 00	0.23000000D 02	0.98466004D-03	0.31644109D-01
508	257	0.11947065D 00	0.24000000D 02	0.93968230D-03	0.30221559D-01
514	263	0.11947065D 00	0.25000000D 02	0.89861989D-03	0.28920912D-01
520	269	0.11947065D 00	0.26000000D 02	0.86098400D-03	0.27727187D-01
526	275	0.11947065D 00	0.27000000D 02	0.82636384D-03	0.26627749D-01
532	281	0.11947065D 00	0.28000000D 02	0.79441163D-03	0.25611882D-01
538	287	0.11947065D 00	0.29000000D 02	0.76483103D-03	0.24670425D-01
544	293	0.11947065D 00	0.30000000D 02	0.73736794D-03	0.23795512D-01
550	299	0.11947065D 00	0.31000000D 02	0.71180328D-03	0.22980340D-01
556	305	0.11947065D 00	0.32000000D 02	0.68794711D-03	0.22219009D-01
562	311	0.11947065D 00	0.33000000D 02	0.66563403D-03	0.21506361D-01
568	317	0.11947065D 00	0.34000000D 02	0.64471922D-03	0.20837885D-01
574	323	0.11947065D 00	0.35000000D 02	0.62507549D-03	0.20209601D-01
580	329	0.11947065D 00	0.36000000D 02	0.60659054D-03	0.19617999D-01
586	335	0.11947065D 00	0.37000000D 02	0.58916496D-03	0.19059961D-01
592	341	0.11947065D 00	0.38000000D 02	0.57271033D-03	0.18532717D-01
598	347	0.11947065D 00	0.39000000D 02	0.55714783D-03	0.18033787D-01
604	353	0.11947065D 00	0.40000000D 02	0.54240691D-03	0.17560958D-01
610	359	0.11947065D 00	0.41000000D 02	0.52842430D-03	0.17112233D-01
616	365	0.11947065D 00	0.42000000D 02	0.51514300D-03	0.16685820D-01
622	371	0.11947065D 00	0.43000000D 02	0.50251164D-03	0.16280096D-01
628	377	0.11947065D 00	0.44000000D 02	0.49048368D-03	0.15893595D-01
634	383	0.11947065D 00	0.45000000D 02	0.47901698D-03	0.15524982D-01
640	389	0.11947065D 00	0.46000000D 02	0.46807318D-03	0.15173047D-01
646	395	0.11947065D 00	0.47000000D 02	0.45761735D-03	0.14836683D-01
652	401	0.11947065D 00	0.48000000D 02	0.44761761D-03	0.14514882D-01
658	407	0.11947065D 00	0.49000000D 02	0.43804480D-03	0.14206717D-01
664	413	0.11947065D 00	0.50000000D 02	0.42887216D-03	0.13911343D-01
670	419	0.11947065D 00	0.51000000D 02	0.42007515D-03	0.13627979D-01
676	425	0.11947065D 00	0.52000000D 02	0.41163118D-03	0.13355910D-01
682	431	0.11947065D 00	0.53000000D 02	0.40351945D-03	0.13094472D-01

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
688	437	0.11947065D 00	0.54000000D 02	0.39572074D-03	0.12843056D-01
694	443	0.11947065D 00	0.55000000D 02	0.38821730D-03	0.12601097D-01
700	449	0.11947065D 00	0.56000000D 02	0.38099268D-03	0.12368072D-01
706	455	0.11947065D 00	0.57000000D 02	0.37403164D-03	0.12143495D-01
712	461	0.11947065D 00	0.58000000D 02	0.36732003D-03	0.11926916D-01
718	467	0.11947065D 00	0.59000000D 02	0.36084471D-03	0.11717915D-01
724	473	0.11947065D 00	0.60000000D 02	0.35459339D-03	0.11516103D-01
730	479	0.11947065D 00	0.61000000D 02	0.34855470D-03	0.11321114D-01
736	485	0.11947065D 00	0.62000000D 02	0.34271794D-03	0.11132609D-01
742	491	0.11947065D 00	0.63000000D 02	0.33707319D-03	0.10950269D-01
748	497	0.11947065D 00	0.64000000D 02	0.33161112D-03	0.10773799D-01
754	503	0.11947065D 00	0.65000000D 02	0.32632302D-03	0.10602917D-01
760	509	0.11947065D 00	0.66000000D 02	0.32120071D-03	0.10437366D-01
766	515	0.11947065D 00	0.67000000D 02	0.31623652D-03	0.10276897D-01
772	521	0.11947065D 00	0.68000000D 02	0.31142325D-03	0.10121282D-01
778	527	0.11947065D 00	0.69000000D 02	0.30675412D-03	0.99703024D-02
784	533	0.11947065D 00	0.70000000D 02	0.30222276D-03	0.98237561D-02
790	539	0.11947065D 00	0.71000000D 02	0.29782317D-03	0.96814496D-02
796	545	0.11947065D 00	0.72000000D 02	0.29354968D-03	0.95432025D-02
802	551	0.11947065D 00	0.73000000D 02	0.28939696D-03	0.94088429D-02
808	557	0.11947065D 00	0.74000000D 02	0.28535996D-03	0.92782100D-02
814	563	0.11947065D 00	0.75000000D 02	0.28143391D-03	0.91511503D-02
820	569	0.11947065D 00	0.76000000D 02	0.27761430D-03	0.90275200D-02
826	575	0.11947065D 00	0.77000000D 02	0.27389688D-03	0.89071814D-02
832	581	0.11947065D 00	0.78000000D 02	0.27027758D-03	0.87900056D-02
838	587	0.11947065D 00	0.79000000D 02	0.26675259D-03	0.86758691D-02
844	593	0.11947065D 00	0.80000000D 02	0.26331826D-03	0.85646557D-02
850	599	0.11947065D 00	0.81000000D 02	0.25997115D-03	0.84562540D-02
856	605	0.11947065D 00	0.82000000D 02	0.25670797D-03	0.83505595D-02
862	611	0.11947065D 00	0.83000000D 02	0.25352562D-03	0.82474715D-02
868	617	0.11947065D 00	0.84000000D 02	0.25042112D-03	0.81468953D-02
874	623	0.11947065D 00	0.85000000D 02	0.24739166D-03	0.80487398D-02
880	629	0.11947065D 00	0.86000000D 02	0.24443454D-03	0.79529192D-02
886	635	0.11947065D 00	0.87000000D 02	0.24154722D-03	0.78593508D-02
892	641	0.11947065D 00	0.88000000D 02	0.23872725D-03	0.77679565D-02
898	647	0.11947065D 00	0.89000000D 02	0.23597229D-03	0.76786611D-02
904	653	0.11947065D 00	0.90000000D 02	0.23328014D-03	0.75913936D-02
910	659	0.11947065D 00	0.91000000D 02	0.23064867D-03	0.75060852D-02
916	665	0.11947065D 00	0.92000000D 02	0.22807585D-03	0.74226713D-02
922	671	0.11947065D 00	0.93000000D 02	0.22555974D-03	0.73410889D-02
928	677	0.11947065D 00	0.94000000D 02	0.22309849D-03	0.72612790D-02
934	683	0.11947065D 00	0.95000000D 02	0.22069032D-03	0.71831840D-02
940	689	0.11947065D 00	0.96000000D 02	0.21833355D-03	0.71067496D-02
946	695	0.11947065D 00	0.97000000D 02	0.21602653D-03	0.70319230D-02
952	701	0.11947065D 00	0.98000000D 02	0.21376772D-03	0.69586545D-02
958	707	0.11947065D 00	0.99000000D 02	0.21155561D-03	0.68868957D-02
964	713	0.11947065D 00	0.10000000D 03	0.20938878D-03	0.68166005D-02

TOTAL NUMBER OF FUNCTION EVALUATIONS 965

```
PROBLEM 3.5 THE THERMAL DECOMPOSITION OF OZONE


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  2
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000D 01
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.1000000D 03
EXACT INTERVALS WILL BE PROVIDED
ADAM'S PREDICTOR CORRECTOR OR OPTION 2 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.10000D 01    0.0



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.4203509D-02

THE ORDER OF THE INTEGRATION IS  4



FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.9119119D-02

THE ORDER OF THE INTEGRATION IS  3

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.1097182D-01


****************************************
*METHOD SELECTED IS GEAR'S STIFF OPTION   2*
****************************************
```

CE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
288	106	0.25966726D-01	0.10000000D 01	0.15990569D 00	0.85020257D 00
331	119	0.39307504D-01	0.20000000D 01	0.38701923D-01	0.59694343D 00
357	129	0.12062752D 00	0.30000000D 01	0.16204240D-01	0.38166053D 00
382	137	0.15304123D 00	0.40000000D 01	0.95463053D-02	0.26011142D 00
398	143	0.11947065D 00	0.50000000D 01	0.66564850D-02	0.19262848D 00
414	149	0.11947065D 00	0.60000000D 01	0.50833259D-02	0.15172434D 00
426	155	0.11947065D 00	0.70000000D 01	0.41019794D-02	0.12472977D 00
437	161	0.11947065D 00	0.80000000D 01	0.34339446D-02	0.10571180D 00
446	167	0.11947065D 00	0.90000000D 01	0.29508519D-02	0.91638801D-01
454	173	0.11947065D 00	0.10000000D 02	0.25857070D-02	0.80825064D-01
460	179	0.11947065D 00	0.11000000D 02	0.23002484D-02	0.72266019D-01
466	185	0.11947065D 00	0.12000000D 02	0.20710859D-02	0.65328592D-01
472	191	0.11947065D 00	0.13000000D 02	0.18831369D-02	0.59594917D-01
480	197	0.11947065D 00	0.14000000D 02	0.17262475D-02	0.54778592D-01
486	203	0.11947065D 00	0.15000000D 02	0.15933363D-02	0.50676961D-01
494	209	0.11947065D 00	0.16000000D 02	0.14793165D-02	0.47142722D-01
502	215	0.11947065D 00	0.17000000D 02	0.13804415D-02	0.44066289D-01
510	221	0.11947065D 00	0.18000000D 02	0.12938916D-02	0.41364496D-01
518	227	0.11947065D 00	0.19000000D 02	0.12175045D-02	0.38973099D-01
526	233	0.11947065D 00	0.20000000D 02	0.11495946D-02	0.36841712D-01
534	239	0.11947065D 00	0.21000000D 02	0.10888289D-02	0.34930259D-01
542	245	0.11947065D 00	0.22000000D 02	0.10341392D-02	0.33206480D-01
550	251	0.11947065D 00	0.23000000D 02	0.98466004D-03	0.31644109D-01
558	257	0.11947065D 00	0.24000000D 02	0.93968230D-03	0.30221559D-01
566	263	0.11947065D 00	0.25000000D 02	0.89801989D-03	0.28920912D-01
574	269	0.11947065D 00	0.26000000D 02	0.86098400D-03	0.27727187D-01
582	275	0.11947065D 00	0.27000000D 02	0.82636384D-03	0.26627749D-01
590	281	0.11947065D 00	0.28000000D 02	0.79441163D-03	0.25611882D-01
598	287	0.11947065D 00	0.29000000D 02	0.76483103D-03	0.24670425D-01
606	293	0.11947065D 00	0.30000000D 02	0.73736794D-03	0.23795512D-01
614	299	0.11947065D 00	0.31000000D 02	0.71180328D-03	0.22980340D-01
622	305	0.11947065D 00	0.32000000D 02	0.68794711D-03	0.22219009D-01
630	311	0.11947065D 00	0.33000000D 02	0.66563403D-03	0.21506361D-01
638	317	0.11947065D 00	0.34000000D 02	0.64471922D-03	0.20837885D-01
646	323	0.11947065D 00	0.35000000D 02	0.62507549D-03	0.20209601D-01
654	329	0.11947065D 00	0.36000000D 02	0.60659054D-03	0.19617999D-01
662	335	0.11947065D 00	0.37000000D 02	0.58916496D-03	0.19059961D-01
670	341	0.11947065D 00	0.38000000D 02	0.57271033D-03	0.18532717D-01
678	347	0.11947065D 00	0.39000000D 02	0.55714783D-03	0.18033787D-01
686	353	0.11947065D 00	0.40000000D 02	0.54240691D-03	0.17560958D-01
694	359	0.11947065D 00	0.41000000D 02	0.52842430D-03	0.17112233D-01
702	365	0.11947065D 00	0.42000000D 02	0.51514300D-03	0.16658820D-01
710	371	0.11947065D 00	0.43000000D 02	0.50251164D-03	0.16280096D-01
718	377	0.11947065D 00	0.44000000D 02	0.49048368D-03	0.15893595D-01
726	383	0.11947065D 00	0.45000000D 02	0.47901698D-03	0.15524982D-01
734	389	0.11947065D 00	0.46000000D 02	0.46807318D-03	0.15173047D-01
742	395	0.11947065D 00	0.47000000D 02	0.45761735D-03	0.14836683D-01
750	401	0.11947065D 00	0.48000000D 02	0.44761761D-03	0.14514882D-01
758	407	0.11947065D 00	0.49000000D 02	0.43804480D-03	0.14206717D-01
766	413	0.11947065D 00	0.50000000D 02	0.42887216D-03	0.13911343D-01
774	419	0.11947065D 00	0.51000000D 02	0.42007515D-03	0.13627979D-01
782	425	0.11947065D 00	0.52000000D 02	0.41163118D-03	0.13355910D-01
790	431	0.11947065D 00	0.53000000D 02	0.40351945D-03	0.13094472D-01

DE. EV.	ACCUM. STEPS	STEP SIZE	INDEP. VARIABLE	DEP. VAR(1)	DEP. VAR(2) ....
798	437	0.11947065D 00	0.54000000D 02	0.39572074D-03	0.12843056D-01
806	443	0.11947065D 00	0.55000000D 02	0.38821730D-03	0.12601097D-01
814	449	0.11947065D 00	0.56000000D 02	0.38099268D-03	0.12368072D-01
822	455	0.11947065D 00	0.57000000D 02	0.37403164D-03	0.12143495D-01
830	461	0.11947065D 00	0.58000000D 02	0.36732003D-03	0.11926916D-01
838	467	0.11947065D 00	0.59000000D 02	0.36084471D-03	0.11717915D-01
846	473	0.11947065D 00	0.60000000D 02	0.35459339D-03	0.11516103D-01
854	479	0.11947065D 00	0.61000000D 02	0.34855470D-03	0.11321114D-01
862	485	0.11947065D 00	0.62000000D 02	0.34271794D-03	0.11132609D-01
870	491	0.11947065D 00	0.63000000D 02	0.33707319D-03	0.10950269D-01
878	497	0.11947065D 00	0.64000000D 02	0.33161112D-03	0.10773799D-01
886	503	0.11947065D 00	0.65000000D 02	0.32632302D-03	0.10602917D-01
894	509	0.11947065D 00	0.66000000D 02	0.32120071D-03	0.10437366D-01
902	515	0.11947065D 00	0.67000000D 02	0.31623652D-03	0.10276897D-01
910	521	0.11947065D 00	0.68000000D 02	0.31142325D-03	0.10121282D-01
918	527	0.11947065D 00	0.69000000D 02	0.30675412D-03	0.99703024D-02
926	533	0.11947065D 00	0.70000000D 02	0.30222276D-03	0.98237561D-02
934	539	0.11947065D 00	0.71000000D 02	0.29782317D-03	0.96814496D-02
942	545	0.11947065D 00	0.72000000D 02	0.29354968D-03	0.95432025D-02
950	551	0.11947065D 00	0.73000000D 02	0.28939696D-03	0.94088429D-02
958	557	0.11947065D 00	0.74000000D 02	0.28535996D-03	0.92782100D-02
966	563	0.11947065D 00	0.75000000D 02	0.28143391D-03	0.91511503D-02
974	569	0.11947065D 00	0.76000000D 02	0.27761430D-03	0.90275200D-02
982	575	0.11947065D 00	0.77000000D 02	0.27389688D-03	0.89071814D-02
990	581	0.11947065D 00	0.78000000D 02	0.27027758D-03	0.87900056D-02
998	587	0.11947065D 00	0.79000000D 02	0.26675259D-03	0.86758691D-02
1006	593	0.11947065D 00	0.80000000D 02	0.26331826D-03	0.85646557D-02
1014	599	0.11947065D 00	0.81000000D 02	0.25997115D-03	0.84562540D-02
1022	605	0.11947065D 00	0.82000000D 02	0.25670797D-03	0.83505595D-02
1030	611	0.11947065D 00	0.83000000D 02	0.25352562D-03	0.82474715D-02
1038	617	0.11947065D 00	0.84000000D 02	0.25042112D-03	0.81468953D-02
1046	623	0.11947065D 00	0.85000000D 02	0.24739166D-03	0.80487398D-02
1054	629	0.11947065D 00	0.86000000D 02	0.24443454D-03	0.79529192D-02
1062	635	0.11947065D 00	0.87000000D 02	0.24154722D-03	0.78593508D-02
1070	641	0.11947065D 00	0.88000000D 02	0.23872725D-03	0.77679565D-02
1078	647	0.11947065D 00	0.89000000D 02	0.23597229D-03	0.76786611D-02
1086	653	0.11947065D 00	0.90000000D 02	0.23328014D-03	0.75913936D-02
1094	659	0.11947065D 00	0.91000000D 02	0.23064867D-03	0.75060852D-02
1102	665	0.11947065D 00	0.92000000D 02	0.22807585D-03	0.74226713D-02
1110	671	0.11947065D 00	0.93000000D 02	0.22555974D-03	0.73410889D-02
1118	677	0.11947065D 00	0.94000000D 02	0.22309849D-03	0.72612790D-02
1126	683	0.11947065D 00	0.95000000D 02	0.22069032D-03	0.71831840D-02
1134	689	0.11947065D 00	0.96000000D 02	0.21833355D-03	0.71067496D-02
1142	695	0.11947065D 00	0.97000000D 02	0.21602653D-03	0.70319230D-02
1150	701	0.11947065D 00	0.98000000D 02	0.21376772D-03	0.69586545D-02
1158	707	0.11947065D 00	0.99000000D 02	0.21155561D-03	0.68868957D-02
1166	713	0.11947065D 00	0.10000000D 03	0.20938878D-03	0.68166005D-02

TOTAL NUMBER OF FUNCTION EVALUATIONS 1167

```
      SUBROUTINE DIFFUN(T,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,1),DY(1)
      COMMON NCFNS
      NCFNS = NCFNS + 1
      TK = 0.0006D00*DEXP(20.7D00 - 15.D+3/Y(1,2))
      DY(4) = 1.29613589D00*(Y(1,2) - Y(1,4)) + 1.03690872D+4*TK*Y(1,1)
      DY(1) = 1866.76D00*(Y(1,3) - Y(1,1)*(1.D00 + TK))
      DY(2) =  1752.D00 - 269.267D00*Y(1,2) + 266.667D00*Y(1,4)
      DY(3) = 0.1D00 + 320.D00*Y(1,1) - 321.D00*Y(1,3)
      RETURN
      END
```

```
      SUBROUTINE PEDERV( T,Y,PW,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(8,4),PW(4,4)
      TK = 0.0006D00*DEXP(20.7D00 - 15.D+3/Y(1,2))
      DTKY2 = TK*(15.D+3/(Y(1,2)**2))
      PW(1,1) = - 1866.76E00*(1.E00 + TK)
      PW(1,2) = - 1866.76E00*Y(1,1)*DTKY2
      PW(1,3) = 1866.76E00
      PW(1,4) = 0.E00
      PW(2,1) = 0.E00
      PW(2,2) = - 269.267E00
      PW(2,3) = 0.E00
      PW(2,4) = 266.667E00
      PW(3,1) = 320.E00
      PW(3,2) = 0.E00
      PW(3,3) = - 321.E00
      PW(3,4) = 0.E00
      PW(4,1) = 1.03690872E+4*TK
      PW(4,2) = 1.296135899E00 + 1.03690872E+4*Y(1,1)*DTKY2
      PW(4,3) = 0.E00
      PW(4,4) = - 1.296135899E00
      RETURN
      END
```

```
PROBLEM 3.6  THE TRANSIENT BEHAVIOR OF A CATALYTIC FLUIDIZED BED


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  4
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000 01
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.5000000D 02
EXACT INTERVALS WILL BE PROVIDED
ADAMS  PREDICTOR CORRECTOR OR OPTION 1 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.0            0.6000D 03    0.1000D 00    0.7592D 03



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS    0.3429429D-03

THE ORDER OF THE INTEGRATION IS  1




FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS    0.4980829D-03

THE ORDER OF THE INTEGRATION IS  1

STEP SIZE CALCULATED FOR EPS =    0.1000000D-03   IS =  0.3429429D 00


****************************************
*METHOD SELECTED IS GEAR'S STIFF OPTION   1*
****************************************
```

NO. EV. ACCUM. STEPS STEP SIZE INDEP. VARIABLE DEP. VAR(1) DEP. VAR(2) ....

212	86	0.65837412D-01	0.10000000D 01	0.72841189D-01	0.75771536D 03	0.72945665D-01	0.75853330D 03
233	93	0.32594315D-01	0.20000000D 01	0.69211174D-01	0.75774065D 03	0.69312777D-01	0.75855864D 03
250	98	0.12069786D 00	0.30000000D 01	0.68180288D-01	0.75773457D 03	0.68281044D-01	0.75855244D 03
260	103	0.20528179D 00	0.40000000D 01	0.67885260D-01	0.75771952D 03	0.67985744D-01	0.75853722D 03
269	108	0.20528179D 00	0.50000000D 01	0.67809504D-01	0.75770172D 03	0.67909887D-01	0.75851924D 03
276	113	0.20528179D 00	0.60000000D 01	0.67795591D-01	0.75768307D 03	0.67895921D-01	0.75850041D 03
281	118	0.20528179D 00	0.70000000D 01	0.67799348D-01	0.75766409D 03	0.67899639D-01	0.75848125D 03
286	123	0.20528179D 00	0.80000000D 01	0.67808212D-01	0.75764495D 03	0.67908467D-01	0.75846192D 03
291	128	0.20528179D 00	0.90000000D 01	0.67818579D-01	0.75762568D 03	0.67918800D-01	0.75844246D 03
296	133	0.20528179D 00	0.10000000D 02	0.67829420D-01	0.75760629D 03	0.67929606D-01	0.75842288D 03
301	138	0.20528179D 00	0.11000000D 02	0.67840440D-01	0.75758679D 03	0.67940591D-01	0.75840319D 03
306	143	0.20528179D 00	0.12000000D 02	0.67851556D-01	0.75756718D 03	0.67951673D-01	0.75838339D 03
311	148	0.20528179D 00	0.13000000D 02	0.67862744D-01	0.75754746D 03	0.67962826D-01	0.75836347D 03
316	153	0.20528179D 00	0.14000000D 02	0.67873998D-01	0.75752762D 03	0.67974044D-01	0.75834344D 03
321	158	0.20528179D 00	0.15000000D 02	0.67885316D-01	0.75750767D 03	0.67985326D-01	0.75832329D 03
326	163	0.20528179D 00	0.16000000D 02	0.67896697D-01	0.75748760D 03	0.67996672D-01	0.75830303D 03
331	168	0.20528179D 00	0.17000000D 02	0.67908142D-01	0.75746742D 03	0.68008081D-01	0.75828265D 03
336	173	0.20528179D 00	0.18000000D 02	0.67919651D-01	0.75744712D 03	0.68019554D-01	0.75826215D 03
341	178	0.20528179D 00	0.19000000D 02	0.67931225D-01	0.75742671D 03	0.68031092D-01	0.75824154D 03
346	183	0.20528179D 00	0.20000000D 02	0.67942863D-01	0.75740618D 03	0.68042693D-01	0.75822081D 03
351	188	0.20528179D 00	0.21000000D 02	0.67954566D-01	0.75738553D 03	0.68054360D-01	0.75819996D 03
356	193	0.20528179D 00	0.22000000D 02	0.67966335D-01	0.75736476D 03	0.68066092D-01	0.75817899D 03
361	198	0.20528179D 00	0.23000000D 02	0.67978169D-01	0.75734388D 03	0.68077889D-01	0.75815790D 03
366	203	0.20528179D 00	0.24000000D 02	0.67990070D-01	0.75732288D 03	0.68089752D-01	0.75813669D 03
371	208	0.20528179D 00	0.25000000D 02	0.68002036D-01	0.75730175D 03	0.68101681D-01	0.75811536D 03
376	213	0.20528179D 00	0.26000000D 02	0.68014069D-01	0.75728051D 03	0.68113677D-01	0.75809391D 03
381	218	0.20528179D 00	0.27000000D 02	0.68026169D-01	0.75725914D 03	0.68125739D-01	0.75807234D 03

DE. EV. ACCUM. STEPS STEP SIZE INDEP. VARIABLE DEP. VAR(1) DEP. VAR(2) ....

386	223	0.20528179D 00	0.28000000D 02	0.68038337D-01	0.75723765D 03	0.68137868D-01	0.75805064D 03
391	228	0.20528179D 00	0.29000000D 02	0.68050572D-01	0.75721604D 03	0.68150065D-01	0.75802881D 03
396	233	0.20528179D 00	0.30000000D 02	0.68062875D-01	0.75719431D 03	0.68162329D-01	0.75800687D 03
401	238	0.20528179D 00	0.31000000D 02	0.68075246D-01	0.75717245D 03	0.68174662D-01	0.75798480D 03
406	243	0.20528179D 00	0.32000000D 02	0.68087686D-01	0.75715047D 03	0.68187063D-01	0.75796260D 03
411	248	0.20528179D 00	0.33000000D 02	0.68100195D-01	0.75712836D 03	0.68199533D-01	0.75794028D 03
416	253	0.20528179D 00	0.34000000D 02	0.68112773D-01	0.75710613D 03	0.68212071D-01	0.75791783D 03
421	258	0.20528179D 00	0.35000000D 02	0.68125421D-01	0.75708377D 03	0.68224680D-01	0.75789525D 03
426	263	0.20528179D 00	0.36000000D 02	0.68138139D-01	0.75706128D 03	0.68237357D-01	0.75787254D 03
431	268	0.20528179D 00	0.37000000D 02	0.68150927D-01	0.75703866D 03	0.68250105D-01	0.75784971D 03
436	273	0.20528179D 00	0.38000000D 02	0.68163786D-01	0.75701592D 03	0.68262924D-01	0.75782674D 03
441	278	0.20528179D 00	0.39000000D 02	0.68176715D-01	0.75699305D 03	0.68275813D-01	0.75780365D 03
446	283	0.20528179D 00	0.40000000D 02	0.68189716D-01	0.75697005D 03	0.68288773D-01	0.75778042D 03
451	288	0.20528179D 00	0.41000000D 02	0.68202789D-01	0.75694692D 03	0.68301805D-01	0.75775706D 03
456	293	0.20528179D 00	0.42000000D 02	0.68215933D-01	0.75692365D 03	0.68314908D-01	0.75773357D 03
461	298	0.20528179D 00	0.43000000D 02	0.68229150D-01	0.75690026D 03	0.68328084D-01	0.75770995D 03
466	303	0.20528179D 00	0.44000000D 02	0.68242440D-01	0.75687673D 03	0.68341332D-01	0.75768619D 03
471	308	0.20528179D 00	0.45000000D 02	0.68255803D-01	0.75685307D 03	0.68354653D-01	0.75766230D 03
476	313	0.20528179D 00	0.46000000D 02	0.68269239D-01	0.75682928D 03	0.68368047D-01	0.75763827D 03
481	318	0.20528179D 00	0.47000000D 02	0.68282749D-01	0.75680535D 03	0.68381514D-01	0.75761411D 03
486	323	0.20528179D 00	0.48000000D 02	0.68296332D-01	0.75678128D 03	0.68395055D-01	0.75758981D 03
491	328	0.20528179D 00	0.49000000D 02	0.68309991D-01	0.75675708D 03	0.68408671D-01	0.75756537D 03
496	333	0.20528179D 00	0.50000000D 02	0.68323724D-01	0.75673275D 03	0.68422361D-01	0.75754080D 03

TOTAL NUMBER OF FUNCTION EVALUATIONS 497

```
PROBLEM 3.6  THE TRANSIENT BEHAVIOR OF A CATALYTIC FLUIDIZED BED


OUTPUT AT INTERVALS OF  0.10D 01
NUMBER OF EQUATIONS IS  4
INITIAL STEP SIZE IS  0.10D-03
MINIMUM STEP SIZE IS  0.10D-09
TOLERANCE IS  0.10D-04
MAXIMUM DERIVATIVE TO BE USED IS 6
INITIAL VALUE OF THE INDEPENDENT VARIABLE IS  0.0
MAXIMUM STEP SIZE ALLOWED IS    0.1000000D 01
SUPERIOR LIMIT OF THIS INTEGRATION IS    0.5000000D 02
EXACT INTERVALS WILL BE PROVIDED
ADAM'S PREDICTOR CORRECTOR OR OPTION 2 OF GEAR'S STIFF METHOD WILL BE SELECTED

THE INITIAL VALUES OF THE DEPENDENT VARIABLES ARE
   0.0           0.60000D 03    0.10000D 00    0.75920D 03



FOR EPS =     0.1000000D-06
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.3429429D-03

THE ORDER OF THE INTEGRATION IS  1




FOR EPS =     0.1000000D-03
ADAM'S PREDICTOR CORRECTOR AVERAGE STEP SIZE IS   0.4980829D-03

THE ORDER OF THE INTEGRATION IS  1

STEP SIZE CALCULATED FOR EPS =   0.1000000D-03   IS =  0.3429429D 00


********************************************
*METHOD SELECTED IS GEAR'S STIFF OPTION   2*
********************************************
```

DE. EV. ACCUM. STEPS STEP SIZE INDEP. VARIABLE DEP. VAR(1) DEP. VAR(2) ....

268	86	0.65837412D-01	0.10000000D 01	0.72841189D-01	0.75771536D 03	0.72945665D-01	0.75853330D 03
297	93	0.32606437D-01	0.20000000D 01	0.69211174D-01	0.75774065D 03	0.69312777D-01	0.75855864D 03
318	98	0.12055691D 00	0.30000000D 01	0.68180284D-01	0.75773457D 03	0.68281041D-01	0.75855244D 03
328	103	0.20506897D 00	0.40000000D 01	0.67885255D-01	0.75771952D 03	0.67985740D-01	0.75853722D 03
337	108	0.20506897D 00	0.50000000D 01	0.67809503D-01	0.75770172D 03	0.67909886D-01	0.75851924D 03
344	113	0.20506897D 00	0.60000000D 01	0.67795591D-01	0.75768307D 03	0.67895921D-01	0.75850041D 03
349	118	0.20506897D 00	0.70000000D 01	0.67799348D-01	0.75766409D 03	0.67899639D-01	0.75848125D 03
354	123	0.20506897D 00	0.80000000D 01	0.67808212D-01	0.75764495D 03	0.67908467D-01	0.75846192D 03
359	128	0.20506897D 00	0.90000000D 01	0.67818579D-01	0.75762568D 03	0.67918800D-01	0.75844246D 03
364	133	0.20506897D 00	0.10000000D 02	0.67829420D-01	0.75760629D 03	0.67929606D-01	0.75842288D 03
369	138	0.20506897D 00	0.11000000D 02	0.67840440D-01	0.75758679D 03	0.67940591D-01	0.75840319D 03
374	143	0.20506897D 00	0.12000000D 02	0.67851556D-01	0.75756718D 03	0.67951673D-01	0.75838339D 03
379	148	0.20506897D 00	0.13000000D 02	0.67862744D-01	0.75754746D 03	0.67962826D-01	0.75836347D 03
384	153	0.20506897D 00	0.14000000D 02	0.67873998D-01	0.75752762D 03	0.67974044D-01	0.75834344D 03
389	158	0.20506897D 00	0.15000000D 02	0.67885316D-01	0.75750767D 03	0.67985326D-01	0.75832329D 03
394	163	0.20506897D 00	0.16000000D 02	0.67896697D-01	0.75748760D 03	0.67996672D-01	0.75830303D 03
399	168	0.20506897D 00	0.17000000D 02	0.67908142D-01	0.75746742D 03	0.68008081D-01	0.75828265D 03
404	173	0.20506897D 00	0.18000000D 02	0.67919651D-01	0.75744712D 03	0.68019554D-01	0.75826215D 03
409	178	0.20506897D 00	0.19000000D 02	0.67931225D-01	0.75742671D 03	0.68031092D-01	0.75824154D 03
414	183	0.20506897D 00	0.20000000D 02	0.67942863D-01	0.75740618D 03	0.68042693D-01	0.75822081D 03
419	188	0.20506897D 00	0.21000000D 02	0.67954566D-01	0.75738553D 03	0.68054360D-01	0.75819996D 03
424	193	0.20506897D 00	0.22000000D 02	0.67966335D-01	0.75736476D 03	0.68066092D-01	0.75817899D 03
429	198	0.20506897D 00	0.23000000D 02	0.67978169D-01	0.75734388D 03	0.68077889D-01	0.75815790D 03
434	203	0.20506897D 00	0.24000000D 02	0.67990070D-01	0.75732288D 03	0.68089752D-01	0.75813669D 03
439	208	0.20506897D 00	0.25000000D 02	0.68002036D-01	0.75730175D 03	0.68101681D-01	0.75811536D 03
444	213	0.20506897D 00	0.26000000D 02	0.68014069D-01	0.75728051D 03	0.68113677D-01	0.75809391D 03
449	218	0.20506897D 00	0.27000000D 02	0.68026170D-01	0.75725914D 03	0.68125739D-01	0.75807234D 03

DE. EV. ACCUM. STEPS   STEP SIZE   INDEP. VARIABLE DEP. VAR(1) DEP. VAR(2) ....

454	223	0.20506897D 00	0.28000000D 02	0.68038337D-01	0.757237650D 03	0.681378680D-01	0.75805064D 03
459	228	0.20506897D 00	0.29000000D 02	0.68050572D-01	0.75721604D 03	0.68150065D-01	0.75802881D 03
464	233	0.20506897D 00	0.30000000D 02	0.68062875D-01	0.75719431D 03	0.68162329D-01	0.75800687D 03
469	238	0.20506897D 00	0.31000000D 02	0.68075246D-01	0.75717245D 03	0.68174662D-01	0.75798480D 03
474	243	0.20506897D 00	0.32000000D 02	0.68087686D-01	0.75715047D 03	0.68187063D-01	0.75796260D 03
479	248	0.20506897D 00	0.33000000D 02	0.68100195D-01	0.75712836D 03	0.68199533D-01	0.75794028D 03
484	253	0.20506897D 00	0.34000000D 02	0.68112773D-01	0.75710613D 03	0.68212071D-01	0.75791783D 03
489	258	0.20506897D 00	0.35000000D 02	0.68125421D-01	0.75708377D 03	0.68224680D-01	0.75789525D 03
494	263	0.20506897D 00	0.36000000D 02	0.68138139D-01	0.75706128D 03	0.68237357D-01	0.75787254D 03
499	268	0.20506897D 00	0.37000000D 02	0.68150927D-01	0.75703866D 03	0.68250105D-01	0.75784971D 03
504	273	0.20506897D 00	0.38000000D 02	0.68163786D-01	0.75701592D 03	0.68262924D-01	0.75782674D 03
509	278	0.20506897D 00	0.39000000D 02	0.68176715D-01	0.75699305D 03	0.68275813D-01	0.75780365D 03
514	283	0.20506897D 00	0.40000000D 02	0.68189716D-01	0.75697005D 03	0.68288773D-01	0.75778042D 03
519	288	0.20506897D 00	0.41000000D 02	0.68202789D-01	0.75694692D 03	0.68301805D-01	0.75775706D 03
524	293	0.20506897D 00	0.42000000D 02	0.68215934D-01	0.75692365D 03	0.68314908D-01	0.75773357D 03
529	298	0.20506897D 00	0.43000000D 02	0.68229150D-01	0.75690026D 03	0.68328084D-01	0.75770995D 03
534	303	0.20506897D 00	0.44000000D 02	0.68242440D-01	0.75687673D 03	0.68341332D-01	0.75768619D 03
539	308	0.20506897D 00	0.45000000D 02	0.68255803D-01	0.75685307D 03	0.68354653D-01	0.75766230D 03
544	313	0.20506897D 00	0.46000000D 02	0.68269239D-01	0.75682928D 03	0.68368047D-01	0.75763827D 03
549	318	0.20506897D 00	0.47000000D 02	0.68282749D-01	0.75680535D 03	0.68381514D-01	0.75761411D 03
554	323	0.20506897D 00	0.48000000D 02	0.68296332D-01	0.75678128D 03	0.68395055D-01	0.75758981D 03
559	328	0.20506897D 00	0.49000000D 02	0.68309991D-01	0.75675708D 03	0.68408671D-01	0.75756537D 03
564	333	0.20506897D 00	0.50000000D 02	0.68323724D-01	0.75673275D 03	0.68422361D-01	0.75754080D 03

TOTAL NUMBER OF FUNCTION EVALUATIONS   565

```
      C     --------------------------------------------------------------------
      C     --------------------------------------------------------------------
      C
            IMPLICIT REAL*8(A-H,O-Z)
            REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMO2,MTMCO2,KCSHM
            REAL*4 PRIRA
      C
            COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2
            COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
            COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
            COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCC3SM,HSM,
           1PCCM,HCC3CM,HCM,PSOM,PSOMO,PCOM,PCOMO,PN2M,PN2MO,PSNM,PSNMC,PCNM,
           2PCNMC
            COMMON/BLOCK5/YM(8,11)
            COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,DSCOM,
           1   CM,VMO,CMMAX,DPSNM,DSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,
           2   DSPHM,DCSBM,KCSHM,FM
            COMMON/BLOCK8/FFTIM
            COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL
            COMMON/BLOK10/J
            COMMON/BLOK11/T
            COMMON/BLCK12/DELT
            COMMON NOFNS
      C
            EXTERNAL DIFFUN,PEDERV,OUTP
      C
      C     --------------------------------------------------------------------
      C
      C
            WRITE(6,610)
        610 FORMAT(1H1,/1X,36('*')/
           |               1X,'*PROBLEM 3.7     A BIOLOGICAL SYSTEM*'/
           | 1X,36('*')//)
            CALL DATA
      C
            TIME=0.0D00
            SEL = 1.D00
            FTIME=16.0D00
            FFTIM=2.05D00
            DELT=0.05D00
            PRIRA = SEL/DELT
            KCS = 0
            T=0.0D00
            SPLM=0.0D00
      C
      C
      C     --------------------------------------------------------------------
      C
      C
          5 IF(SPLM .EQ. 0.0D00) CALL INTAL
            SPLM=SPLM+DELT
            IF(SPLM .GT. FTIME) GO TO 40
            IF(DABS(SPLM-FFTIM) .LE. 0.0001D00) GO TO 2
            GO TO 1
```

```
            C                                                             ETSS 500
 0034       2      CONTINUE                                               ETSS 510
 0035              CALL FORCE                                             ETSS 520
            C                                                             ETSS 530
 0036       1      CONTINUE                                               ETSS 540
 0037              IF(KOS .LT. IFIX(4.E00*PRIRA))GO TO 11
 0038              KOS = 0
 0039              WRITE(6,215)
 0040        215   FORMAT(1H1///)
 0041         11   CONTINUE
 0042              KOS = KOS + 1
            C                                                             ETSS 550
            C      -------------------------------------------------------ETSS 560
            C                                                             ETSS 570
 0043              CALL MUSCLE                                            ETSS 580
            C                                                             ETSS 590
            C      -------------------------------------------------------ETSS 600
            C                                                             ETSS 610
 0044              GO TO 5                                                ETSS 620
            C                                                             ETSS 630
 0045         40   CONTINUE                                               ETSS 640
            C
 0046              STOP                                                   ETSS 700
 0047              END                                                    ETSS 710
```

```
0001            SUBROUTINE MUSCLE
          C
          C     ---------------------------------------------------------------------
          C
          C
0002            IMPLICIT REAL*8(A-H,O-Z)
0003            REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMC2,MTMCC2,KCSHM,MTMMO2
0004            REAL*8 INTAMA,INTAMM,MM,MMS,MMC
0005            REAL*8 L1,L2
0006            REAL*4 PW,PPW
          C
          C     ---------------------------------------------------------------------
          C
0007            COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2
0008            COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
0009            COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
0010            COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PC2MO,PSCM,HCO3SM,HSM,
               1PCCM,HCO3CM,HCM,PSOM,PSOMC,PCCM,PCCMC,PN2M,PN2MC,PSNM,PSNMC,PCNM,
               2PCNMC
0011            COMMON/BLOCK5/YM(8,11)
0012            COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,DSCOM,
               1   CM,VMB,CMMAX,DPSNM,DSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,
               2   DSPHM,DCSBM,KCSHM,FM
0013            COMMON/BLOCK8/FFTIM
0014            COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL
0015            COMMON/BLOK10/J
0016            COMMON/BLOK11/T
0017            COMMON/BLOK12/DELT
0018            COMMON NOFNS
          C
0019            DIMENSION SAVE(12,11),YMAX(11),YF(11),ERROR(11),PW(121),
               1          PPW(121),LLL(11),MMM(11)
          C
0020            EXTERNAL DIFFUN,PEDERV,OUTP
          C
          C     ---------------------------------------------------------------------
          C
          C
0021            IF(T     .EQ. 0.0000) GO TO 111
0022            GO TO 222
0023      111   NOFNS=0
0024            J=0
0025            JSTART=0
          C
          C        PARAMETERS FOR GEAR ' S  SYSTEM
          C
0026            FSEL=SEL
0027            N=11
          C
          C     PARTIAL DERIVATIVES PROVIDED ANALYTICALLY IN PEDERV
          C
0028            MF = 1
          C
0029            HMIN=1.0D-10
```

```
0030            EPS = 5.0D-05
0031            MAXDER=6
0032            HMAX=DELT
0033            DSEL=DELT
0034            H=1.0D-04
0035            DELTA = DSEL
0036            DO 10 I=1,11
0037            YMAX(I)=1.0D00
0038      10    CONTINUE
0039      222   CONTINUE
          C
0040            IF(T .GT. 0.0D00) H=DELT
0041            IF(DABS(T-FFTIM) .LE. DELT) H=1.0D-04
          C
          C    ---------------------------------------------------------------------
          C
0042            CALL MPGRM2(N,T,YM,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG,
               1JSTART,MAXDER,PW,PPW,LLL,MMM,YF,DELTA,DIFFUN,PEDERV,OUTP)
          C
          C    ---------------------------------------------------------------------
          C
          C    ---------------------------------------------------------------------
          C
0043      100   CONTINUE
          C
0044            RETURN
0045            END
```

```
      FUNCTION SLOPE1(CMAX,ALPHAO,PO2,PCO2)                             SLOP   10
C                                                                       SLOP   20
C     ------------------------------------------------------------------SLOP   30
C                                                                       SLOP   40
      IMPLICIT REAL*8(A-H,O-Z)                                          SLOP   50
C                                                                       SLOP   60
C     ------------------------------------------------------------------SLOP   70
C                                                                       SLOP   80
C                                                                       SLOP   90
      R=0.004273D00+0.04326D00*PCO2**(-0.535D00)                        SLOP  100
      V=R*PO2                                                           SLOP  110
      A=0.925D00+5.6D00*V+90.0D00*V*V                                   SLOP  120
      U=0.925D00*V+2.8D00*V*V+30.00*V*V*V                               SLOP  130
      SLOPE1=CMAX*A*R/(1.0D00+U)**2+ALPHAO                              SLOP  140
C                                                                       SLOP  150
C     ------------------------------------------------------------------SLOP  160
C                                                                       SLOP  170
      RETURN                                                            SLOP  180
      END                                                               SLOP  190
```

```
      FUNCTION SLOPE2(CMMAX,ALPHA0,PC2)                                 SLOP  10
C                                                                       SLOP  20
C     ------------------------------------------------------------------SLOP  30
C                                                                       SLOP  40
      IMPLICIT REAL*8(A-H,O-Z)                                          SLOP  50
C                                                                       SLOP  60
C     ------------------------------------------------------------------SLOP  70
C                                                                       SLOP  80
      CC=99.509D00                                                      SLOP  90
      AA=-321.221D00                                                    SLOP 100
      BB=3.228D00                                                       SLOP 110
      SLOPE2=-CMMAX*AA/(BB+PC2)**2+ALPHA0                               SLOP 120
C                                                                       SLOP 130
C     ------------------------------------------------------------------SLOP 140
C                                                                       SLOP 150
      RETURN                                                            SLOP 160
      END                                                               SLOP 170
```

```
      FUNCTION SLOPE3(CMAX,ALPHAC,PC2,PCC2)                             SLOP  10
C                                                                       SLOP  20
C     ------------------------------------------------------------------SLOP  30
C                                                                       SLOP  40
      IMPLICIT REAL*8(A-H,O-Z)                                          SLOP  50
C                                                                       SLOP  60
C     ------------------------------------------------------------------SLOP  70
C                                                                       SLOP  80
C                                                                       SLOP  90
      R=0.004273D00+0.04326D00*PCC2**(-0.535D00)                        SLOP 100
      V=R*PC2                                                           SLOP 110
      A=0.925D00+5.6D00*V+90.0D00*V*V                                   SLOP 120
      U=0.925D00*V+2.8D00*V*V+30.00*V*V*V                               SLOP 130
      SLOPE3=R*CMAX*((5.6D00+180.0D00*V)*R/(1.0D00+U)**2-2.0D00*A*A*R/  SLOP 140
     1             (1.0D00+U)**3)                                       SLOP 150
C                                                                       SLOP 160
C     ------------------------------------------------------------------SLOP 170
C                                                                       SLOP 180
      RETURN                                                            SLOP 190
      END                                                               SLOP 200
```

```
      FUNCTION SLOPE4(CMMAX,PO2)                                        SLOP  10
C                                                                       SLOP  20
C     ------------------------------------------------------------------SLOP  30
C                                                                       SLOP  40
      IMPLICIT REAL*8(A-H,O-Z)                                          SLOP  50
C                                                                       SLOP  60
C     ------------------------------------------------------------------SLOP  70
C                                                                       SLOP  80
C                                                                       SLOP  90
      AA=-321.221D00                                                    SLOP 100
      BB=3.228D00                                                       SLOP 110
      CC=99.509D00                                                      SLOP 120
      SLOPE4=2.0D00*CMMAX*AA/(BB+PO2)**3                                SLOP 130
C                                                                       SLOP 140
C     ------------------------------------------------------------------SLOP 150
C                                                                       SLOP 160
      RETURN                                                            SLOP 170
      END                                                               SLOP 180
```

```
      SUBROUTINE CONC(CMAX,PO2,PCO2,CO2,CCO2)                           CONC  10
C                                                                       CONC  20
C     ******************************************************************CONC  30
C     *                                                                *CONC  40
C     *   THIS ROUTINE CALCULATES THE CONTENTS OF OXYGEN AND CARBON -   *CONC  50
C     *   DIOXIDE CORROSPONDING TO GIVEN PARTIAL PRESSURES OF OXYGEN    *CONC  60
C     *   AND CARBON DIOXIDE FOR WHOLE BLOOD                           *CONC  70
C     *                                                                *CONC  80
C     ******************************************************************CONC  90
      IMPLICIT REAL*8(A-H,O-Z)                                          CONC 100
      DATA SOLO2/3.0D-05/                                               CONC 110
      V=(0.004273+0.04326*PCO2**(-0.535))*PO2                           CONC 120
      U=0.925*V+2.8*V**2+30.*V**3                                       CONC 130
      CO2=CMAX*U/(1.+U)+SOLO2*PO2                                       CONC 140
      CCO2=(0.149-0.014*U/(1.+U))*PCO2**0.35                            CONC 150
      RETURN                                                            CONC 160
      END                                                               CONC 170
```

```
      SUBROUTINE CUTP(N,T,J,YF,SAVE,H,DT,NOFNS)                         CUTP  10
C                                                                       CUTP  20
C    -------------------------------------------------------------------CUTP  30
C                                                                       CUTP  40
      IMPLICIT REAL*8(A-H,O-Z)                                          CUTP  50
C                                                                       CUTP  60
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCO3SM,HSM, CUTP  70
     1PCCM,HCO3CM,HCM,PSCM,PSCMC,PCCM,PCCMC,PN2M,PN2MC,PSNM,PSNMC,PCNM, CUTP  80
     2PCNMO                                                             CUTP  90
C                                                                       CUTP 100
      DIMENSION YF(1),SAVE(12,01)                                       CUTP 110
C                                                                       CUTP 120
C    -------------------------------------------------------------------CUTP 130
C                                                                       CUTP 140
C                                                                       CUTP 150
C                                                                       CUTP 160
      WRITE(6,8)                                                        CUTP 170
 8    FORMAT(1X,125('-')/)                                              CUTP 180
C                                                                       CUTP 190
      WRITE(6,5)                                                        CUTP 200
 5    FORMAT(1X,'NUMERICAL INTEGRATION SOL.  (GEAR)  :')                CUTP 210
      WRITE(6,1)T,DT,NOFNS,J,H                                          CUTP 220
 1    FORMAT(//1X,'TIME =',F10.3,5X,'STEP SIZE =',D12.5,5X,'DE. EV. =', CUTP 230
     1         I6,5X,'ACCUM. STEPS =',I6,5X,' H  =',D12.5/)             CUTP 240
C                                                                       CUTP 250
      WRITE(6,2)(YF(I),I=1,8)                                           CUTP 260
 2    FORMAT(22X,'PCO2M =',D20.12,5X,'HCO3PM =',D20.12//                CUTP 270
     1      22X,'PSCM  =',D20.12,5X,'HCO3SM =',D20.12,5X,'HSM =',D20.12CUTP 280
     2      //22X,'PCCM  =',D20.12,5X,'HCO3CM =',D20.12,5X,'HCM =',     CUTP 290
     3      D20.12/)                                                    CUTP 300
C                                                                       CUTP 310
C                                                                       CUTP 320
      WRITE(6,22)(YF(I),I=9,11)                                         CUTP 330
 22   FORMAT(22X,'PO2M =',D20.12,5X,'PSCM =',D20.12, 5X,'PCOM =',D20.12)CUTP 340
C                                                                       CUTP 350
```

```
C                                                                       CUTP 360
C                                                                       CUTP 370
C     ------------------------------------------------------------------CUTP 380
C                                                                       CUTP 390
      RETURN                                                            CUTP 400
      END                                                               CUTP 410
```

```
      SUBROUTINE INTAL                                                  INTA  10
C                                                                       INTA  20
C     ------------------------------------------------------------------INTA  30
C                                                                       INTA  40
      IMPLICIT REAL*8(A-H,O-Z)                                          INTA  50
      REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMO2,MTMCO2,KCSHM              INTA  60
C                                                                       INTA  70
      COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2 INTA  80
      COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO                          INTA  90
      COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM              INTA 100
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCO3SM,HSM, INTA 110
     1PCOM,HCO3OM,HOM,PSOM,PSOMO,PCOM,PCOMO,PN2M,PN2MO,PSNM,PSNMO,PONM, INTA 120
     2PONMO                                                             INTA 130
      COMMON/BLOCK5/YM(8,11)                                            INTA 140
      COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSCM,DSCCM,      INTA 150
     1   QM,VMB,CMMAX,DPSNM,DSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,    INTA 160
     2   DSPHM,DCSBM,KCSHM,FM                                           INTA 170
C                                                                       INTA 180
C                                                                       INTA 190
C     ------------------------------------------------------------------INTA 200
C                                                                       INTA 210
      PCO2M=45.1110066333D00                                            INTA 220
      HCO3PM=0.0247542034915D00                                         INTA 230
      CARBM=0.00238165321520D00                                         INTA 240
      PO2M=37.0169310044D00                                             INTA 250
C                                                                       INTA 260
      CALL CONC(CMAX,PO2M,PCO2M,CO2M,CCO2M)                             INTA 270
C                                                                       INTA 280
C     ------------------------------------------------------------------INTA 290
C                                                                       INTA 300
      DPSCM=MTMO2/5.0D00                                                INTA 310
      DSCCM=MTMO2/15.0D00                                               INTA 320
C                                                                       INTA 330
      DPSNM=DPSCM/2.0D00                                                INTA 340
      DSCNM=DSCCM/2.0D00                                                INTA 350
```

```
C                                                                       INTA 360
C                                                                       INTA 370
            PO2MO=PO2M                                                  INTA 380
            PN2M=PN2AM                                                  INTA 390
            PN2MO=PN2M                                                  INTA 400
C                                                                       INTA 410
            PSOM=PO2M-MTMO2/DPSOM                                       INTA 420
            PCOM=PSOM-MTMO2/DSCOM                                       INTA 430
C                                                                       INTA 440
            PSOMO=PSOM                                                  INTA 450
            PCOMO=PCOM                                                  INTA 460
            PSNM=PN2M                                                   INTA 470
            PCNM=PN2M                                                   INTA 480
            PSNMO=PSNM                                                  INTA 490
            PCNMO=PCNM                                                  INTA 500
C                                                                       INTA 510
C                                                                       INTA 520
C     ------------------------------------------------------------------INTA 530
C                                                                       INTA 540
C                                                                       INTA 550
            DELSPM=2.0D00                                               INTA 560
            DELCSM= 5.0D00                                              INTA 570
            XM=0.9D00                                                   INTA 580
            FM=0.80D00                                                  INTA 590
            TCSM=0.5D00                                                 INTA 600
            RCSM=29.882D00                                              INTA 610
            RSPM=0.95D00                                                INTA 620
            ZM=0.10D00                                                  INTA 630
C                                                                       INTA 640
C                                                                       INTA 650
C     ------------------------------------------------------------------INTA 660
C     ------------------------------------------------------------------INTA 670
C                                                                       INTA 680
C                                                                       INTA 690
            PSOM=PCO2M+DELSPM                                           INTA 700
            DSPCM=MTMCO2*(XM-ZM)/DELSPM                                 INTA 710
```

```
      HCO3SM=0.027D00
      HSM=(KU*ALPHAC*PSCM-MTMCO2*ZM/VSM)/(KV/K*HCO3SM)
      CSPBM=MTMCO2*(1.0D00+ZM-XM)/(RSPM*HCO3SM-HCO3PM)
      CSPHM=MTMCO2*(1.0D00+ZM-XM)/(HSM/RSPM-ALPHAC*KPRIME*PCO2M/HCO3PM)
C
C     ------------------------------------------------------------------
C
      PCCM=PSCM+DELCSM
      DCSCM=MTMCO2/DELCSM*XM
      HCO3CM=0.012D00
      HCM=(KU*ALPHAC*PCCM-MTMCO2*(FM-XM)/VCM)/(KV/K*HCO3CM)
      DCSBM=MTMCO2*(1.0D00-XM)/(RCSM*HCO3CM-HCO3SM)
      KCSHM=MTMCO2*(1.0D00-XM)/(TCSM*HCM-HSM)
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C
C
      YM(1,1)=PCO2M
      YM(1,2)=HCO3PM
      YM(1,3)=PSCM
      YM(1,4)=HCO3SM
      YM(1,5)=HSM
      YM(1,6)=PCCM
      YM(1,7)=HCO3CM
      YM(1,8)=HCM
      YM(1,9)=PO2M
      YM(1,10)=PSCM
      YM(1,11)=PCOM
C
C
      RETURN
      END
```

```
      SUBROUTINE FORCE                                                  FORC  10
C                                                                       FORC  20
C     ------------------------------------------------------------------FORC  30
C                                                                       FORC  40
      IMPLICIT REAL*8(A-H,O-Z)                                          FORC  50
C                                                                       FORC  60
      COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM              FORC  70
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSOM,HCO3SM,HSM, FORC  80
     1PCOM,HCO3CM,HCM,PSCM,PSOMO,PCOM,PCOMO,PN2M,PN2MO,PSNM,PSNMO,PCNM, FORC  90
     2PCNMO                                                             FORC 100
C                                                                       FORC 110
C     ------------------------------------------------------------------FORC 120
C                                                                       FORC 130
      CMAX=0.201000                                                     FORC 140
      PCO2AM=PCO2M                                                      FORC 150
      HCO3AM=HCO3PM                                                     FORC 160
      CARBAM=CARBM                                                      FORC 170
      PO2AM=60.0000                                                     FORC 180
      CALL CONC(CMAX,PO2AM,PCO2AM,CO2AM,CCO2AM)                         FORC 190
C                                                                       FORC 200
C     ------------------------------------------------------------------FORC 210
C                                                                       FORC 220
      RETURN                                                            FORC 230
      END                                                               FORC 240
```

```
      SUBROUTINE PEDERV(T,YM,PW,M)                                      PEDE  10
C                                                                       PEDE  20
C     ------------------------------------------------------------------PEDE  30
C                                                                       PEDE  40
      IMPLICIT REAL*8(A-H,O-Z)                                          PEDE  50
      REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMO2,MTMCO2,KCSHM,MTMMO2       PEDE  60
      REAL*4 PW                                                         PEDE  70
C                                                                       PEDE  80
      COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAC,HB,ABN2,ATN2 PEDE  90
      COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO                          PEDE 100
      COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM              PEDE 110
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCO3SM,HSM, PEDE 120
     1PCCM,HCO3CM,HCM,PSCM,PSOMC,PCOM,PCCMC,PN2M,PN2MO,PSNM,PSNMO,PCNM, PEDE 130
     2PCNMO                                                             PEDE 140
      COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,CSCCM,      PEDE 150
     1   QM,VMB,CMMAX,DPSNM,CSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,CSPBM,    PEDE 160
     2   DSPHM,DCSBM,KCSHM,FM                                           PEDE 170
C                                                                       PEDE 180
      DIMENSION YM(08,11),PW(11,11)                                     PEDE 190
C                                                                       PEDE 200
C     ------------------------------------------------------------------PEDE 210
C                                                                       PEDE 220
      VPM=VMB*(100.0D00-HCRIT)/100.0D00                                 PEDE 230
      QPM=QM *(100.0D00-HCRIT)/100.0D00                                 PEDE 240
      VEM=VMB*HCRIT/100.0D00                                            PEDE 250
      QEM=QM *HCRIT/100.0D00                                            PEDE 260
C                                                                       PEDE 270
C     ------------------------------------------------------------------PEDE 280
C                                                                       PEDE 290
      TIME=T                                                            PEDE 300
C                                                                       PEDE 310
      PCO2M=YM(1,1)                                                     PEDE 320
      HCO3PM=YM(1,2)                                                    PEDE 330
      PSCM  =YM(1,3)                                                    PEDE 340
      HCO3SM=YM(1,4)                                                    PEDE 350
```

```
      HSM   =YM(1,5)                                                    PECE 360
      PCOM  =YM(1,6)                                                    PECE 370
      HCO3CM=YM(1,7)                                                    PECE 380
      HCM   =YM(1,8)                                                    PECE 390
      PO2M=YM(1,9)                                                      PECE 400
      PSOM=YM(1,10)                                                     PECE 410
      PCOM=YM(1,11)                                                     PECE 420
C                                                                       PECE 430
C                                                                       PECE 440
      IF(PCOM .LT. 0.0000)PCOM=0.0000                                   PECE 450
      IF(PCOM .LE. 10.0000) MTMMO2=MTMO2*PCOM/10.0000                   PECE 460
      IF(PCOM .LE. 10.0000) GO TO 31                                    PECE 470
      MTMMO2=MTMO2                                                      PECE 480
   31 CONTINUE                                                          PECE 490
C                                                                       PECE 500
C     ------------------------------------------------------------------PECE 510
C                                                                       PECE 520
C                                                                       PECE 530
C     ------------------------------------------------------------------PECE 540
C                                                                       PECE 550
C          BLOOD                                                        PECE 560
C                                                                       PECE 570
C     ------------------------------------------------------------------PECE 580
C                                                                       PECE 590
      B1=QPM*ALPHAC*KPRIME/VPM*(PCO2AM/HCO3AM-PCO2M/HCO3PM)             PECE 600
      B2=DSPHM     *(HSM/RSPM-ALPHAC*KPRIME*PCO2M/HCO3PM)               PECE 610
      B3=-2.303000*ALPHAC*KPRIME/(2.000*VPM*BETAP)*(PCO2AM/HCO3AM+PCO2M/PECE 620
     1   HCO3PM)                                                        PECE 630
      B4=-ALPHAC*KPRIME*PCO2M/HCO3PM**2                                 PECE 640
      B5=KPRIME/HCO3PM                                                  PECE 650
      B6=QPM/VPM*(HCO3AM-HCO3PM)+DSPBM/VPM*(RSPM*HCO3SM-HCO3PM)         PECE 660
C                                                                       PECE 670
      SM=(CO2M-ALPHAC*PO2M)/CMAX*100.0000                               PECE 680
C                                                                       PECE 690
      A1=(100.0000-SM)*R*KZ/100.0000                                    PECE 700
      A2=SM*R*KC/100.0000                                               PECE 710
```

```
      A4=KPRIME/(R*KC)                                                  PECE 720
      F=A1*HCO3PM/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)                     PECE 730
     1   +A2*HCO3PM/(R*HCO3PM*KO+ALPHAC*KPRIME*PCO2M)                   PECE 740
C                                                                       PECE 750
      CARBM=F*HB*HCO3PM/(A4+F*HCO3PM)                                   PECE 760
      VERCAB=-QEM*(CARBAM-CARBM)                                        PECE 770
C                                                                       PECE 780
      B7=ALPHAC*QM*(PCO2AM-PCO2M)/VMB+DSPCM*(PSCM-PCO2M)/VMB-VERCAB/VMB PECE 790
      B8=VPM/B4*(B4*B6+B5*B7-B1-B2*B3)                                  PECE 800
      B9=VPM/B4*(B4/VPM-B5/VMB-B3)                                      PECE 810
      B10=-VPM*B5/(B4*VMB)                                              PECE 820
      C1=(QEM/VEM)*R*(HCO3AM-HCO3PM)                                    PECE 830
      C2=(R*B6-C1)*VEM                                                  PECE 840
      C3=R*VEM/VPM                                                      PECE 850
      D1=B1/R                                                           PECE 860
      D2=-2.303D00*ALPHAC*KPRIME/(2.D00*VEM*BETAE*R)*                   PECE 870
     1   (PCO2AM/HCO3AM+PCO2M/HCO3PM)                                   PECE 880
C                                                                       PECE 890
C                                                                       PECE 900
      RO2HB=QM*((CO2M-ALPHAC*PO2M)-(CO2AM-ALPHAC*PO2AM))/22.4D00        PECE 910
C                                                                       PECE 920
      D3=1.5D00*VERCAB+0.6D00*RO2HB                                     PECE 930
      E1=(B1+B2*B3-R*D1-R*D2*D3)/(R*D2)                                 PECE 940
      E2=B3/(R*D2)                                                      PECE 950
      C4=1.D00+(1.D00+C3)*B10                                           PECE 960
      C5=C2-(1.D00+C3)*B8                                               PECE 970
      C6=C3-(1.D00+C3)*B9                                               PECE 980
      F1=C5-E1*C4                                                       PECE 990
      F2=E2*C4-C6                                                       PECE1000
C                                                                       PECE1010
C    -------------------------------------------------------------------PECE1020
C                                                                       PECE1030
      VPRPR=F1/F2                                                       PECE1040
      VERER=E1+E2*F1/F2                                                 PECE1050
      REPB=B8+B9*F1/F2+B10*(E1+E2*F1/F2)                                PECE1060
C                                                                       PECE1070
```

```
C     ---------------------------------------------------------------PEDE1080
C                                                                     PEDE1090
C                                                                     PEDE1100
C                                                                     PEDE1110
C     ---------------------------------------------------------------PEDE1120
C                                                                     PEDE1130
      V=(.004273D00+0.04326D00*PCO2M**(-0.535D00))*PC2M               PEDE1140
      U=0.925D00*V+2.8D00*V*V+30.0D00*V*V*V                           PEDE1150
C                                                                     PEDE1160
      DVDPB=(0.04326D00*(-0.535D00)*PCO2M**(-1.535D00))*PC2M          PEDE1170
      DVDPC2=0.004273D00+0.04326*PCO2M**(-0.535D00)                   PEDE1180
      DUDPB=(0.925D00+5.6D00*V+90.0D00*V*V)*DVDPB                     PEDE1190
      DUDPO2=(0.925D00+5.6D00*V+90.0D00*V*V)*DVDPC2                   PEDE1200
C                                                                     PEDE1210
      A=0.925D00+5.6D00*V+90.0D00*V*V                                 PEDE1220
      DRDPB=0.04326D00*(-0.535D00)*PCO2M**(-1.535D00)                 PEDE1230
      DADPB=(5.6D00+180.0D00*V)*PO2M*DRDPB                            PEDE1240
      DCDPB=CMAX*DUDPB/(1.0D00+U)**2                                  PEDE1250
      DAMBPB=CMAX*(-2.0D00*A*DVDPC2*DUDPB/(1.0D00+U)**3+              PEDE1260
     1        (A*DRDPB+DVDPO2*DADPB)/(1.0D00+U)**2)                   PEDE1270
C                                                                     PEDE1280
      DSDPB= 100.0D00/(1.0D00+U)**2*(0.925D00+5.6D00*V+90.0D00*V*V)*  PEDE1290
     1      (.04326D00*(-0.535D00)*PCO2M**(-1.535D00)*PC2M)           PEDE1300
      DSDPC2=100.0D00*DUDPC2/(1.0D00+U)**2                            PEDE1310
C                                                                     PEDE1320
      DA1DPB=-DSDPB*R*KZ/100.0D00                                     PEDE1330
      DA1DPO=-DSDPC2*R*KZ/100.0D00                                    PEDE1340
C                                                                     PEDE1350
      DA2DPO=DSDPC2*R*KO/100.0D00                                     PEDE1360
      DA2DPB=DSDPB*R*KO/100.0D00                                      PEDE1370
C                                                                     PEDE1380
      DGDF=A4*H8*HCO3PM/(A4+F*HCO3PM)**2                              PEDE1390
C                                                                     PEDE1400
      DFDPB=-A1*HCO3PM*ALPHAC*KPRIME/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)  PEDE1410
     1      **2-A2*HCO3PM*ALPHAC*KPRIME/(R*HCO3PM*KO+ALPHAC*KPRIME*   PEDE1420
     2      PCO2M)**2+HCO3PM*DA1DPB/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)  PEDE1430
```

```
     3                +HCO3PM*DA2DPB/(R*HCO3PM*KO+ALPHAC*KPRIME*PCO2M)     PECE1440
      DFDPC2=HCO3PM*DA1DPO/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)              PECE1450
     1        +HCO3PM*DA2DPC/(R*HCO3PM*KO+ALPHAC*KPRIME*PCO2M)             PECE1460
C                                                                          PECE1470
      DCBDPB=DGDF*DFDPB                                                    PECE1480
      DCBDPC=DGDF*DFDPC2                                                   PECE1490
C                                                                          PECE1500
C     -----------------------------------------------------------------PECE1510
C                                                                          PECE1520
      DGDBP=A4*HB*F/(A4+F*HCO3PM)**2                                       PECE1530
      DFDBP=ALPHAC*KPRIME*PCO2M*(A1/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)**2PECE1540
     1       +A2/(R*HCO3PM*KO+ALPHAC*KPRIME*PCO2M)**2)                     PECE1550
C                                                                          PECE1560
      DCBDBP=DGDBP+DGDF*DFDBP                                              PECE1570
C                                                                          PECE1580
C                                                                          PECE1590
C     -----------------------------------------------------------------PECE1600
C                                                                          PECE1610
      DVCBPB=QEM*DCBDPB                                                    PECE1620
      DVCBBP=QEM*DCBDBP                                                    PECE1630
      DVCBDO=QEM*DCBDPO                                                    PECE1640
C                                                                          PECE1650
C                                                                          PECE1660
C     -----------------------------------------------------------------PECE1670
C                                                                          PECE1680
C                                                                          PECE1690
      DRHBPB=QM*CMAX*DSDPB/(22.4D00*100.0D00)                              PECE1700
      DRHBDO=QM*CMAX*DSDPO2/(22.4D00*100.0D00)                             PECE1710
C                                                                          PECE1720
C                                                                          PECE1730
C     -----------------------------------------------------------------PECE1740
C                                                                          PECE1750
      DB1DPB=-QPM*ALPHAC*KPRIME/(VPM*HCO3PM)                               PECE1760
      DB1DBP=QPM*ALPHAC*KPRIME/VPM*PCO2M/HCO3PM**2                         PECE1770
C                                                                          PECE1780
      DB2DPB=-ALPHAC*KPRIME*DSPHM/HCO3PM                                   PECE1790
```

```
      DB2DBP=DSPHM*ALPHAC*KPRIME*PCO2M/(HCO3PM*HCO3PM)                  PEDE1800
      DB2DHS=DSPHM/RSPM                                                 PEDE1810
C                                                                       PEDE1820
      DB3DPB=-2.303D00*ALPHAC*KPRIME/(2.D000*VPM*BETAP*HCO3PM)          PEDE1830
      DB3DBP=2.303D00*ALPHAC*KPRIME/(2.D000*VPM*BETAP)*PCO2M/(HCO3PM*   PEDE1840
     1       HCO3PM)                                                    PEDE1850
C                                                                       PEDE1860
      DB4DPB=-KPRIME*ALPHAC/(HCO3PM*HCO3PM)                             PEDE1870
      DB4DBP=2.D000*ALPHAC*KPRIME*PCO2M/(HCO3PM*HCO3PM*HCO3PM)          PEDE1880
C                                                                       PEDE1890
      DB5DBP=-KPRIME/(HCO3PM*HCO3PM)                                    PEDE1900
C                                                                       PEDE1910
      DB6DBP=-(QPM+DSPBM)/VPM                                           PEDE1920
      DB6DBS=DSPBM*RSPM/VPM                                             PEDE1930
C                                                                       PEDE1940
      DB7DPB=-ALPHAC*QM/VMB-DSPCM/VMB-DVCBPB/VMB                        PEDE1950
      DB7DBP=-DVCBBP/VMB                                                PEDE1960
      DB7DPS=DSPCM/VMB                                                  PEDE1970
      DB7DPC=-DVCBDC/VMB                                                PEDE1980
C                                                                       PEDE1990
      DB8DPB=VPM/B4*(B6*DB4DPB+B5*DB7DPB-DB1DPB-B2*DB3DPB-B3*DB2DPB)    PEDE2000
     1       -(B4*B6+B5*B7-B1-B2*B3)*VPM/(B4*B4)*DB4DPB                 PEDE2010
      DB8DBP=VPM/B4*(B6*DB4DBP+B4*DB6DBP+B5*DB7DBP+B7*DB5DBP-DB1DBP     PEDE2020
     1       -B2*DB3DBP-B3*DB2DBP)-(B4*B6+B5*B7-B1-B2*B3)*VPM/(B4*B4)*  PEDE2030
     1       DB4DBP                                                     PEDE2040
      DB8DPS=VPM/B4*B5*DB7DPS                                           PEDE2050
      DB8DBS=VPM*DB6DBS                                                 PEDE2060
      DB8DHS=-VPM/B4*B3*DB2DHS                                          PEDE2070
      DB8DPC=VPM*B5*DB7DPC/B4                                           PEDE2080
C                                                                       PEDE2090
      DB9DPB=VPM/B4*(DB4DPB/VPM-DB3DPB)-VPM/(B4*B4)*(B4/VPM-B5/VMB-B3)  PEDE2100
     1       *DB4DPB                                                    PEDE2110
      DB9DBP=VPM/B4*(DB4DBP/VPM-DB5DBP/VMB-DB3DBP)-VPM/(B4*B4)*         PEDE2120
     1       (B4/VPM-B5/VMB-B3)*DB4DBP                                  PEDE2130
C                                                                       PEDE2140
      DB10PB=VPM/VMB*B5/(B4*B4)*DB4DPB                                  PEDE2150
```

```
      DB1DBP=-VPM/VMB*(B4*DB5DBP-B5*DB4DBP)/(B4*B4)                     PECE2160
C                                                                       PECE2170
C                                                                       PECE2180
      DC1DBP=-R*CEM/VEM                                                 PECE2190
C                                                                       PECE2200
      DC2DBP=VEM*(R*DB6DBP-DC1DBP)                                      PECE2210
      DC2DBS=R*VEM*DB6DBS                                               PECE2220
C                                                                       PECE2230
      DD1DPB=DB1DPB/R                                                   PECE2240
      DD1DBP=DB1DBP/R                                                   PECE2250
C                                                                       PECE2260
      DD2DPB=-2.303D00*ALPHAC*KPRIME/(2.0D00*VEM*BETAE*R*HCO3PM)        PECE2270
      DD2DBP=2.303D00*ALPHAC*KPRIME/(2.0D00*VEM*R*BETAE)*PCO2M/         PECE2280
     1         (HCO3PM*HCO3PM)                                          PECE2290
C                                                                       PECE2300
      DD3DPB=1.5D00*DVCBPB+0.6D00*DRHBPB                                PECE2310
      DD3DBP=1.5D00*DVCBBP                                              PECE2320
      DD3DPO=1.5D00*DVCBDO+0.6D00*DRHBDO                                PECE2330
C                                                                       PECE2340
      DE1DPB=((DB1DPB+B2*DB3DPB+B3*DB2DPB-R*DD1DPB-R*D2*DD3DPB-R*D3*    PECE2350
     1          DD2DPB)*D2-(B1+B2*B3-R*D1-R*D2*D3)*DD2DPB)/(R*D2*D2)    PECE2360
      DE1DBP=((DB1DBP+B2*DB3DBP+B3*DB2DBP-R*DD1DBP-R*D2*DD3DBP-R*D3*    PECE2370
     1          DD2DBP)*D2-(B1+B2*B3-R*D1-R*D2*D3)*DD2DBP)/(R*D2*D2)    PECE2380
      DE1DHS=B3*DB2DHS/(R*D2)                                           PECE2390
      DE1DPO=-DD3DPO                                                    PECE2400
C                                                                       PECE2410
      DE2DPB=(D2*DB3DPB-B3*DD2DPB)/(R*D2*D2)                            PECE2420
      DE2DBP=(D2*DB3DBP-B3*DD2DBP)/(R*D2*D2)                            PECE2430
C                                                                       PECE2440
      DC4DPB=(1.0D00+C3)*DB1DPB                                         PECE2450
      DC4DBP=(1.0D00+C3)*DB1DBP                                         PECE2460
C                                                                       PECE2470
      DC5DPB=-(1.0D00+C3)*DB8DPB                                        PECE2480
      DC5DBP=DC2DBP-(1.0D00+C3)*DB8DBP                                  PECE2490
      DC5DPS=-(1.0D00+C3)*DB8DPS                                        PECE2500
      DC5DBS=DC2DBS-(1.0D00+C3)*DB8DBS                                  PECE2510
```

```
      DC5DHS=-(1.0D00+C3)*DB8DHS                                        PEDE2520
      DC5DPO=-(1.0D00+C3)*DB8DPO                                        PEDE2530
C                                                                       PEDE2540
      DC6DPB=-(1.0D00+C3)*DB9DPB                                        PEDE2550
      DC6DBP=-(1.0D00+C3)*DB9DBP                                        PEDE2560
C                                                                       PEDE2570
C                                                                       PEDE2580
      DF1DPB=DC5DPB-E1*DC4DPB-C4*DE1DPB                                 PEDE2590
      DF1DBP=DC5DBP-E1*DC4DBP-C4*DE1DBP                                 PEDE2600
      DF1DPS=DC5DPS                                                     PEDE2610
      DF1DBS=DC5DBS                                                     PEDE2620
      DF1DHS=DC5DHS-C4*DE1DHS                                           PEDE2630
      DF1DPO=DC5DPO-C4*DE1DPO                                           PEDE2640
C                                                                       PEDE2650
      DF2DPB=E2*DC4DPB+C4*DE2DPB-DC6DPB                                 PEDE2660
      DF2DBP=E2*DC4DBP+C4*DE2DBP-DC6DBP                                 PEDE2670
C                                                                       PEDE2680
C                                                                       PEDE2690
C     ------------------------------------------------------------------PEDE2700
C                                                                       PEDE2710
      DVPRPB=(F2*DF1DPB-F1*DF2DPB)/(F2*F2)                              PEDE2720
      DVPRBP=(F2*DF1DBP-F1*DF2DBP)/(F2*F2)                              PEDE2730
      DVPRPS=DF1DPS/F2                                                  PEDE2740
      DVPRBS=DF1DBS/F2                                                  PEDE2750
      DVPRHS=DF1DHS/F2                                                  PEDE2760
      DVPRDO=DF1DPO/F2                                                  PEDE2770
C                                                                       PEDE2780
      DVERPB=DE1DPB+E2*DVPRPB+VPRPR*DE2DPB                              PEDE2790
      DVERBP=DE1DBP+E2*DVPRBP+VPRPR*DE2DBP                              PEDE2800
      DVERPS=E2*DVPRPS                                                  PEDE2810
      DVERBS=E2*DVPRBS                                                  PEDE2820
      DVERHS=DE1DHS+E2*DVPRHS                                           PEDE2830
      DVERDO=DE1DPO+E2*DVPRDO                                           PEDE2840
C                                                                       PEDE2850
      DREPPB=DB8DPB+B9*DVPRPB+VPRPR*DB9DPB+B10*DVERPB+VERER*DB10PB      PEDE2860
      DREPBP=DB8DBP+B9*DVPRBP+VPRPR*DB9DBP+B10*DVERBP+VERER*DB10BP      PEDE2870
```

```
      DREPPS=DB8DPS+B9*DVPRPS+B10*DVERPS                                PECE2880
      DREPBS=DB8DBS+B9*DVPRBS+B10*DVERBS                                PECE2890
      DREPHS=DB8DHS+B9*DVPRHS+B10*DVERHS                                PECE2900
      DREPDO=DB8DPO+B9*DVPRDO+B10*DVERDO                                PECE2910
C                                                                       PECE2920
C                                                                       PECE2930
C     ------------------------------------------------------------------PECE2940
C                                                                       PECE2950
C                                                                       PECE2960
      AMBC2 =SLOPE1(CMAX,ALPHAC,PC2M,PCC2M)                             PECE2970
      AMBBO2=SLOPE3(CMAX,ALPHAO,PO2M,PCO2M)                             PECE2980
      CALL CONC(CMAX,PO2M,PCO2M,CO2M,CCO2M)                             PECE2990
      AMCC2=SLOPE2(CMMAX,ALPHAC,PCOM)                                   PECE3000
      AMCCO2=SLOPE4(CMMAX,PCOM)                                         PECE3010
C                                                                       PECE3020
C     ------------------------------------------------------------------PECE3030
C                                                                       PECE3040
C                                                                       PECE3050
      PW(1,1)=(-ALPHAC*QM-DSPCM-DVPRPB-DVERPB-DVCBPB)/(ALPHAC*VMB)      PECE3060
      PW(1,2)=(-DVPRBP-DVERBP-DVCBBP)/(ALPHAC*VMB)                      PECE3070
      PW(1,3)=(-DVPRPS-DVERPS+DSPCM)/(ALPHAC*VMB)                       PECE3080
      PW(1,4)=(-DVPRBS-DVERBS)/(ALPHAC*VMB)                             PECE3090
      PW(1,5)=(-DVPRHS-DVERHS)/(ALPHAC*VMB)                             PECE3100
      PW(1,6)=0.0D00                                                    PECE3110
      PW(1,7)=0.0D00                                                    PECE3120
      PW(1,8)=0.0D00                                                    PECE3130
      PW(1,9)=(-DVPRDO-DVERDO-DVCBDO)/(ALPHAC*VMB)                      PECE3140
C                                                                       PECE3150
C     ------------------------------------------------------------------PECE3160
C                                                                       PECE3170
      PW(2,1)=(-DREPPB+DVPRPB)/VPM                                      PECE3180
      PW(2,2)=(-QPM-DSPBM-DREPBP+DVPRBP)/VPM                            PECE3190
      PW(2,3)=(-DREPPS+DVPRPS)/VPM                                      PECE3200
      PW(2,4)=(-DREPBS+DVPRBS+DSPBM*RSPM)/VPM                           PECE3210
      PW(2,5)=(-DREPHS+DVPRHS)/VPM                                      PECE3220
      PW(2,6)=0.0D00                                                    PECE3230
```

```
      PW(2,7)=0.0D00                                                    PECE3240
      PW(2,8)=0.0D00                                                    PECE3250
      PW(2,9)=(-DREPCO+DVPRCO)/VPM                                      PECE3260
C                                                                       PECE3270
C    -------------------------------------------------------------------PECE3280
C                                                                       PECE3290
      PW(3,1)=DSPCM/(ALPHAC*VSM)                                        PECE3300
      PW(3,2)=0.0D00                                                    PECE3310
      PW(3,3)=(-DSPCM-DCSCM-VSM*KU*ALPHAC)/(ALPHAC*VSM)                 PECE3320
      PW(3,4)=KV/K*HSM/ALPHAC                                           PECE3330
      PW(3,5)=KV/K*HCO3SM/ALPHAC                                        PECE3340
      PW(3,6)=DCSCM/(ALPHAC*VSM)                                        PECE3350
      PW(3,7)=0.0D00                                                    PECE3360
      PW(3,8)=0.0D00                                                    PECE3370
C                                                                       PECE3380
C    -------------------------------------------------------------------PECE3390
C                                                                       PECE3400
      PW(4,1)=0.0D00                                                    PECE3410
      PW(4,2)=DSPBM/VSM                                                 PECE3420
      PW(4,3)=KL*ALPHAC                                                 PECE3430
      PW(4,4)=(-DCSBM-DSPBM*RSPM-VSM*KV/K*HSM)/VSM                      PECE3440
      PW(4,5)=-KV/K*HCO3SM                                              PECE3450
      PW(4,6)=0.0D00                                                    PECE3460
      PW(4,7)=DCSBM*RCSM/VSM                                            PECE3470
      PW(4,8)=0.0D00                                                    PECE3480
C                                                                       PECE3490
C    -------------------------------------------------------------------PECE3500
C                                                                       PECE3510
      PW(5,1)=(DSPHM*ALPHAC*KPRIME/HCO3PM)*(-2.303D00*HSM)/(BETASM*VSM) PECE3520
      PW(5,2)=(DSPHM*ALPHAC*KPRIME*PCO2M/HCO3PM**2)*(2.303D00*HSM)/(VSM*PECE3530
     1        BETASM)                                                   PECE3540
      PW(5,3)=-KL*ALPHAC*2.303D00*HSM/BETASM                            PECE3550
      PW(5,4)=KV/K*HSM*2.303D00*HSM/BETASM                              PECE3560
      PW(5,5)=(-KCSHM-DSPHM/RSPM-VSM*KV/K*HCO3SM)*(-2.303D00*HSM)       PECE3570
     1        /(BETASM*VSM)+(KCSHM*(TCSM*HCM-HSM)-DSPHM*(HSM/RSPM-      PECE3580
     2        ALPHAC*KPRIME*PCO2M/HCO3PM)+VSM*(KU*ALPHAC*PSCM-KV/K*HSM*PECE3590
```

```
     3           HCO3SM))*(-2.303D00/(BETASM*VSM))                      PECE3600
      PW(5,6)=0.0D00                                                     PECE3610
      PW(5,7)=0.0D00                                                     PECE3620
      PW(5,8)=KCSHM*TCSM/VSM*(-2.303D00*HSM/BETASM)                      PECE3630
C                                                                        PECE3640
C     -------------------------------------------------------------------PECE3650
C                                                                        PECE3660
      PW(6,1)=0.0D00                                                     PECE3670
      PW(6,2)=0.0D00                                                     PECE3680
      PW(6,3)=DCSCM/(ALPHAC*VCM)                                         PECE3690
      PW(6,4)=0.0D00                                                     PECE3700
      PW(6,5)=0.0D00                                                     PECE3710
      PW(6,6)=(-DCSCM-VCM*KU*ALPHAC)/(ALPHAC*VCM)                        PECE3720
      PW(6,7)=KV/K*HCM/ALPHAC                                            PECE3730
      PW(6,8)=KV/K*HCO3CM/ALPHAC                                         PECE3740
C                                                                        PECE3750
C     -------------------------------------------------------------------PECE3760
C                                                                        PECE3770
      PW(7,1)=0.0D00                                                     PECE3780
      PW(7,2)=0.0D00                                                     PECE3790
      PW(7,3)=0.0D00                                                     PECE3800
      PW(7,4)=DCSBM/VCM                                                  PECE3810
      PW(7,5)=0.0D00                                                     PECE3820
      PW(7,6)=KU*ALPHAC                                                  PECE3830
      PW(7,7)=(-DCSBM*RCSM-VCM*KV/K*HCM)/VCM                             PECE3840
      PW(7,8)=-KV/K*HCO3CM                                               PECE3850
C                                                                        PECE3860
C     -------------------------------------------------------------------PECE3870
C                                                                        PECE3880
      PW(8,1)=0.0D00                                                     PECE3890
      PW(8,2)=0.0D00                                                     PECE3900
      PW(8,3)=0.0D00                                                     PECE3910
      PW(8,4)=0.0D00                                                     PECE3920
      PW(8,5)=-2.303D00*HCM*KCSHM/(VCM*BETACM)                           PECE3930
      PW(8,6)=-2.303D00*KU*ALPHAC*HCM/BETACM                             PECE3940
      PW(8,7)=2.303D00*KV/K*HCM*HCM/BETACM                               PECE3950
```

```
      PW(8,8)=(-VCM*KV/K*HCO3CM-KCSHM*TCSM)*(-2.303D00*HCM/(VCM*BETACM))PECE3960
     1       +((1.0D00-FM)*MTMCO2+VCM*(KL*ALPHAC*PCCM-KV/K*HCM*HCO3CM)PECE3970
     2       -KCSHM*(TCSM*HCM-HSM))*(-2.303D00/(VCM*BETACM))          PECE3980
C                                                                       PECE3990
C                                                                       PECE4000
C     ------------------------------------------------------------------PECE4010
C                                                                       PECE4020
      PW(9,1)=-QM*DCDPB/(VMB*AMBO2)-QM*(CO2AM-CO2M)*DAMBPB/(VMB*AMBO2*  PECE4030
     1      AMBO2)+DPSOM*(PO2M-PSOM)/(AMBO2*AMBO2*VMB)*DAMBPB           PECE4040
C                                                                       PECE4050
      DO 50 I=3,8                                                       PECE4060
   50 PW(I,9)=0.0D00                                                    PECE4070
C                                                                       PECE4080
      PW(9,9)=-QM/(VMB)      -QM*(CO2AM-CO2M)/(VMB*AMBO2*AMBO2)*AMBBO2  PECE4090
     1      -DPSOM/(VMB*AMBO2)+DPSOM*(PO2M-PSOM)/(AMBO2*AMBO2*VMB)*     PECE4100
     2       AMBBO2                                                     PECE4110
C                                                                       PECE4120
      DO 60 I=2,8                                                       PECE4130
      PW(9,I)=0.0D00                                                    PECE4140
   60 CONTINUE                                                          PECE4150
C                                                                       PECE4160
      PW(9,10)=DPSOM/(AMBO2*VMB)                                        PECE4170
      PW(9,11)=0.0D00                                                   PECE4180
C                                                                       PECE4190
C     ------------------------------------------------------------------PECE4200
C                                                                       PECE4210
      DO 61 I=1,8                                                       PECE4220
      PW(10,I)=0.0D00                                                   PECE4230
   61 CONTINUE                                                          PECE4240
C                                                                       PECE4250
      PW(10,9)=DPSOM/(ALPHAO*VSM)                                       PECE4260
      PW(10,10)=-(DPSOM+DSCOM)/(ALPHAO*VSM)                             PECE4270
      PW(10,11)=DSCOM/(ALPHAO*VSM)                                      PECE4280
C                                                                       PECE4290
C     ------------------------------------------------------------------PECE4300
C                                                                       PECE4310
```

```
      DO 62 I=1,8                                                       PECE4320
   62 PW(11,I)=0.0D00                                                   PECE4330
C                                                                       PECE4340
      PW(11,9)=0.0D00                                                   PECE4350
      PW(11,10)=DSCOM/(AMCO2*VCM)                                       PECE4360
      PW(11,11)=-DSCOM/(AMCO2*VCM)-DSCOM*(PSCM-PCCM)/(VCM*AMCO2*AMCO2)* PECE4370
     1          AMCCO2+MTMMO2/(AMCO2*AMCO2*VCM)*AMCCO2                  PECE4380
C                                                                       PECE4390
C                                                                       PECE4400
C     ------------------------------------------------------------------PECE4410
C                                                                       PECE4420
C     ------------------------------------------------------------------PECE4430
C                                                                       PECE4440
      DO 63 I=1,8                                                       PECE4450
      DO 64 J=10,11                                                     PECE4460
   64 PW(I,J)=0.0D00                                                    PECE4470
   63 CONTINUE                                                          PECE4480
C                                                                       PECE4490
C     ------------------------------------------------------------------PECE4500
C     ------------------------------------------------------------------PECE4510
C                                                                       PECE4520
      RETURN                                                            PECE4530
      END                                                               PECE4540
```

```
      SUBROUTINE DATA                                                   DATA  10
C                                                                       DATA  20
C     ------------------------------------------------------------------DATA  30
C                                                                       DATA  40
      IMPLICIT REAL*8(A-H,O-Z)                                          DATA  50
      REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMC2,MTMCO2,KCSHM              DATA  60
C                                                                       DATA  70
      COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAC,HB,ABN2,ATN2 DATA  80
      COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO                          DATA  90
      COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM              DATA 100
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCO3SM,HSM, DATA 110
     1PCCM,HCO3CM,HCM,PSCM,PSCMO,PCOM,PCOMO,PN2M,PN2MO,PSNM,PSNMO,PCNM, DATA 120
     2PCNMO                                                             DATA 130
      COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSCM,DSCCM,      DATA 140
     1    CM,VMB,CMMAX,DPSNM,CSCNM,MTMC2,MTMCO2,TCSM,RCSM,RSPM,CSPBM,   DATA 150
     2    DSPHM,DCSBM,KCSHM,FM                                          DATA 160
C                                                                       DATA 170
C     ------------------------------------------------------------------DATA 180
C                                                                       DATA 190
C                                                                       DATA 200
      HCRIT=39.0D0                                                      DATA 210
      BETAP=-0.0061D00                                                  DATA 220
      BETAE=-0.056D00                                                   DATA 230
      R=0.7D00                                                          DATA 240
      MTMCO2=0.0425D00/22.4D00                                          DATA 250
      MTMC2=0.05D00                                                     DATA 260
      CMAX=0.201D00                                                     DATA 270
      BETASM=-0.0021D00                                                 DATA 280
      BETACM=-0.0195D00                                                 DATA 290
      VSM=3.46D00                                                       DATA 300
      VCM=26.5D00                                                       DATA 310
      KA=300000.D00                                                     DATA 320
      KC=2.4D-05                                                        DATA 330
      HB=0.02058D00                                                     DATA 340
      KPRIME=10.0D00**(-6.1D00)                                         DATA 350
```

```
      KV=89.0D00*60.0D00                                                DATA 360
      KL=0.130D00*60.0D00                                               DATA 370
      K=KPRIME*KV/KU                                                    DATA 380
      KZ=7.2D-08                                                        DATA 390
      KC=8.4D-09                                                        DATA 400
      ALPHAC=3.0D-05                                                    DATA 410
      CMMAX=1.64D-04                                                    DATA 420
      VMB=1.0D00                                                        DATA 430
      GM=0.84D00                                                        DATA 440
      ALPHAC=3.0D-05                                                    DATA 450
C                                                                       DATA 460
C    -------------------------------------------------------------------DATA 470
C                                                                       DATA 480
      PCO2AM=39.0D00                                                    DATA 490
      PO2AM=100.0D00                                                    DATA 500
      HCO3AM=0.022888D00                                                DATA 510
      CARBAM=0.128542682157D-02                                         DATA 520
C                                                                       DATA 530
      CALL CONC(CMAX,PO2AM,PCO2AM,CO2AM,CCO2AM)                         DATA 540
C                                                                       DATA 550
C    -------------------------------------------------------------------DATA 560
C                                                                       DATA 570
      PATM=760.0D00                                                     DATA 580
      PAH2O=47.0D00                                                     DATA 590
      PN2AM=PATM-PAH2O-PCO2AM-PO2AM                                     DATA 600
      ABN2=1.72D-05                                                     DATA 610
      ATN2=1.49D-05                                                     DATA 620
C                                                                       DATA 630
C    -------------------------------------------------------------------DATA 640
C                                                                       DATA 650
C    -------------------------------------------------------------------DATA 660
C                                                                       DATA 670
      RETURN                                                            DATA 680
      END                                                               DATA 690
```

```
      SUBROUTINE DIFFUN(T,YM,DERYM)                                     DIFF  10
C                                                                       DIFF  20
C     ------------------------------------------------------------------DIFF  30
C                                                                       DIFF  40
      IMPLICIT REAL*8(A-H,O-Z)                                          DIFF  50
      REAL*8 KPRIME,KA,KC,KL,KV,K,KZ,KO,MTMC2,MTMCO2,KCSHM,MTMMC2       DIFF  60
C                                                                       DIFF  70
      COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2 DIFF  80
      COMMON/BLOCK2/KPRIME,KA,KC,KL,KV,K,KZ,KO                          DIFF  90
      COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM              DIFF 100
      COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSOM,HCO3SM,HSM, DIFF 110
     1PCOM,HCO3CM,HCM,PSOM,PSOMO,PCOM,PCOMO,PN2M,PN2MO,PSNM,PSNMO,PONM, DIFF 120
     2PONMO                                                             DIFF 130
      COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,DSCOM,      DIFF 140
     1   QM,VMB,CMMAX,DPSNM,DSCNM,MTMC2,MTMCO2,TCSM,RCSM,RSPM,CSPBM,    DIFF 150
     2   DSPHM,CCSBM,KCSHM,FM                                           DIFF 160
      COMMON/BLOCK8/FFTIM                                               DIFF 170
      COMMON NCFNS                                                      DIFF 180
C                                                                       DIFF 190
      DIMENSION YM(08,11),DERYM(11)                                     DIFF 200
C                                                                       DIFF 210
C     ------------------------------------------------------------------DIFF 220
C                                                                       DIFF 230
      ERR=1.0D-05                                                       DIFF 240
      ERR1=1.0D-10                                                      DIFF 250
      VPM=VMB*(100.0D00-HCRIT)/100.0D00                                 DIFF 260
      QPM=QM *(100.0D00-HCRIT)/100.0D00                                 DIFF 270
      VEM=VMB*HCRIT/100.0D00                                            DIFF 280
      QEM=QM *HCRIT/100.0D00                                            DIFF 290
C                                                                       DIFF 300
C     ------------------------------------------------------------------DIFF 310
C                                                                       DIFF 320
      TIME=T                                                            DIFF 330
C                                                                       DIFF 340
      PCO2M=YM(1,1)                                                     DIFF 350
```

```
      HCO3PM=YM(1,2)
      PSCM  =YM(1,3)
      HCO3SM=YM(1,4)
      HSM   =YM(1,5)
      PCCM  =YM(1,6)
      HCO3CM=YM(1,7)
      HCM   =YM(1,8)
      PO2M  =YM(1,9)
      PSOM=YM(1,10)
      PCOM=YM(1,11)
C
C
C
C     ------------------------------------------------------------------
C
      IF(PCCM .LT. 0.0000) PCCM=0.0000
      IF(PCOM .LE. 10.0000) MTMMO2=MTMO2*PCOM/10.0000
      IF(PCCM .LE. 10.0000) GO TO 31
      MTMMO2=MTMO2
   31 CONTINUE
C
C     ------------------------------------------------------------------
C
C
C
      AMBO2=SLOPE1(CMAX,ALPHAO,PO2M,PCO2M)
      CALL CONC(CMAX,PO2M,PCO2M,CO2M,CCO2M)
      AMCO2=SLOPE2(CMMAX,ALPHAC,PCOM)
C
C
C     ------------------------------------------------------------------
C
C        BLOOD
C
C     ------------------------------------------------------------------
C
```

```
      B1=QPM*ALPHAC*KPRIME/VPM*(PCO2AM/HCO3AM-PCO2M/HCO3PM)             DIFF 720
      B2=DSPHM    *(HSM/RSPM-ALPHAC*KPRIME*PCO2M/HCO3PM)                DIFF 730
      B3=-2.303D00*ALPHAC*KPRIME/(2.D00*VPM*BETAP)*(PCO2AM/HCO3AM+PCO2M/DIFF 740
     1   HCO3PM)                                                        DIFF 750
      B4=-ALPHAC*KPRIME*PCO2M/HCO3PM**2                                 DIFF 760
      B5=KPRIME/HCO3PM                                                  DIFF 770
      B6=QPM/VPM*(HCO3AM-HCO3PM)+DSPBM/VPM*(RSPM*HCO3SM-HCO3PM)         DIFF 780
C                                                                       DIFF 790
      SM=(CO2M-ALPHAO*PO2M)/CMAX*100.D00                                DIFF 800
C                                                                       DIFF 810
      A1=(100.D00-SM)*R*KZ/100.D00                                      DIFF 820
      A2=SM*R*KC/100.D00                                                DIFF 830
      A4=KPRIME/(R*KC)                                                  DIFF 840
C                                                                       DIFF 850
      F=A1*HCO3PM/(R*HCO3PM*KZ+ALPHAC*KPRIME*PCO2M)                     DIFF 860
     1   +A2*HCO3PM/(R*HCO3PM*KC+ALPHAC*KPRIME*PCO2M)                   DIFF 870
C                                                                       DIFF 880
      CARBM=F*HB*HCO3PM/(A4+F*HCO3PM)                                   DIFF 890
      VERCAB=-QEM*(CARBAM-CARBM)                                        DIFF 900
C                                                                       DIFF 910
      B7=ALPHAC*QM*(PCO2AM-PCO2M)/VMB+DSPCM*(PSCM-PCO2M)/VMB-VERCAB/VMB DIFF 920
      B8=VPM/B4*(B4*B6+B5*B7-B1-B2*B3)                                  DIFF 930
      B9=VPM/B4*(B4/VPM-B5/VMB-B3)                                      DIFF 940
      B10=-VPM*B5/(B4*VMB)                                              DIFF 950
      C1=(QEM/VEM)*R*(HCO3AM-HCO3PM)                                    DIFF 960
      C2=(R*B6-C1)*VEM                                                  DIFF 970
      C3=R*VEM/VPM                                                      DIFF 980
      D1=B1/R                                                           DIFF 990
      D2=-2.303D00*ALPHAC*KPRIME/(2.D00*VEM*BETAE*R)*                   DIFF1000
     1   (PCO2AM/HCO3AM+PCO2M/HCO3PM)                                   DIFF1010
C                                                                       DIFF1020
      RO2HB=QM*((CO2M-ALPHAC*PO2M)-(CO2AM-ALPHAO*PO2AM))/22.4D00        DIFF1030
C                                                                       DIFF1040
      D3=1.5D00*VERCAB+0.6D00*RO2HB                                     DIFF1050
      E1=(B1+B2*B3-R*D1-R*D2*D3)/(R*D2)                                 DIFF1060
      E2=B3/(R*D2)                                                      DIFF1070
```

```
      C4=1.D00+(1.D00+C3)*B10                                           DIFF1080
      C5=C2-(1.D00+C3)*B8                                               DIFF1090
      C6=C3-(1.D00+C3)*B9                                               DIFF1100
      F1=C5-E1*C4                                                       DIFF1110
      F2=E2*C4-C6                                                       DIFF1120
C                                                                       DIFF1130
C     ------------------------------------------------------------------DIFF1140
C                                                                       DIFF1150
      VPRPR=F1/F2                                                       DIFF1160
      VERER=E1+E2*F1/F2                                                 DIFF1170
      REPB=B8+B9*F1/F2+B10*(E1+E2*F1/F2)                                DIFF1180
C                                                                       DIFF1190
C     ------------------------------------------------------------------DIFF1200
C                                                                       DIFF1210
C                                                                       DIFF1220
      DERYM(1)=(QM*ALPHAC*(PCO2AM-PCO2M)+DSPCM*(PSCM-PCO2M)-VPRPR-VERER DIFF1230
     1   -VERCAB)/(ALPHAC*VMB)                                          DIFF1240
C                                                                       DIFF1250
C                                                                       DIFF1260
      DERYM(2)=(QPM*(HCO3AM-HCO3PM)-REPB+VPRPR+DSPBM*(RSPM*HCO3SM-HCO3PMDIFF1270
     1   ))/VPM                                                         DIFF1280
C                                                                       DIFF1290
C     ------------------------------------------------------------------DIFF1300
C                                                                       DIFF1310
C                                                                       DIFF1320
C     ------------------------------------------------------------------DIFF1330
C                                                                       DIFF1340
C        INTERSTITIAL FLUID                                             DIFF1350
C                                                                       DIFF1360
C     ------------------------------------------------------------------DIFF1370
C                                                                       DIFF1380
C                                                                       DIFF1390
      RSRM=KU*ALPHAC*PSCM-KV/K*HCO3SM*HSM                               DIFF1400
C                                                                       DIFF1410
      DERYM(3)=(-DSPCM*(PSCM-PCO2M)+DCSCM*(PCCM-PSCM)-VSM*RSRM)/        DIFF1420
     1         (ALPHAC*VSM)                                             DIFF1430
```

```
C                                                                       DIFF1440
      DERYM(4)=(DCSBM*(RCSM*HCO3CM-HCO3SM)-DSPBM*(RSPM*HCO3SM-HCO3PM)   DIFF1450
     1          +VSM*RSRM)/VSM                                          DIFF1460
C                                                                       DIFF1470
      DERYM(5)=(KCSHM*(TCSM*HCM-HSM)-DSPHM*(HSM/RSPM-ALPHAC*KPRIME*     DIFF1480
     1          PCO2M/HCO3PM)+VSM*RSRM)*(-2.303D00*HSM/(VSM*BETASM))    DIFF1490
C                                                                       DIFF1500
C                                                                       DIFF1510
C     ------------------------------------------------------------------DIFF1520
C                                                                       DIFF1530
C         INTRACELLULAR FLUID                                           DIFF1540
C                                                                       DIFF1550
C     ------------------------------------------------------------------DIFF1560
C                                                                       DIFF1570
C                                                                       DIFF1580
      RCRM=KU*ALPHAC*PCOM-KV/K*HCO3CM*HCM                               DIFF1590
C                                                                       DIFF1600
      DERYM(6)=(-DCSOM*(PCOM-PSOM)+MTMCO2*FM-VCM*RCRM)/(ALPHAC*VCM)     DIFF1610
C                                                                       DIFF1620
      DERYM(7)=(-DCSBM*(RCSM*HCO3CM-HCO3SM)+(1.D00-FM)*MTMCO2+VCM*RCRM)/DIFF1630
     1          VCM                                                     DIFF1640
C                                                                       DIFF1650
      DERYM(8)=(-KCSHM*(TCSM*HCM-HSM)+(1.D00-FM)*MTMCO2+VCM*RCRM)*      DIFF1660
     1          (-2.303D00*HCM/(BETACM*VCM))                            DIFF1670
C                                                                       DIFF1680
C                                                                       DIFF1690
C     ------------------------------------------------------------------DIFF1700
C                                                                       DIFF1710
C     ------------------------------------------------------------------DIFF1720
C                                                                       DIFF1730
C                                                                       DIFF1740
C                                                                       DIFF1750
      DERYM(9)=(QM*(CO2AM-CO2M)-DPSOM*(PO2M-PSOM))/(AMBO2*VMB)          DIFF1760
C                                                                       DIFF1770
C                                                                       DIFF1780
      DERYM(10)=(DPSOM*(PO2M-PSOM)-DSCOM*(PSOM-PCOM))/(ALPHAC*VSM)      DIFF1790
```

```
C                                                                       CIFF1800
C                                                                       CIFF1810
      DERYM(11)=(DSCCM*(PSCM-PCCM)-MTMMC2)/(AMCO2*VCM)                  CIFF1820
C                                                                       CIFF1830
C                                                                       CIFF1840
C     ------------------------------------------------------------------CIFF1850
C                                                                       CIFF1860
      IF(TIME .LE. FFTIM) GO TO 20                                      CIFF1870
      GO TO 21                                                          CIFF1880
   20 CONTINUE                                                          CIFF1890
C                                                                       CIFF1900
C     ------------------------------------------------------------------CIFF1910
C                                                                       CIFF1920
      IF(DABS(DERYM(1)*ALPHAC) .LE. ERR) DERYM(1)=0.0D00                CIFF1930
      IF(DABS(DERYM(2)*VPM) .LE. ERR) DERYM(2)=0.0D00                   CIFF1940
      IF(DABS(DERYM(3)*ALPHAC) .LE. ERR) DERYM(3)=0.0D00                CIFF1950
      IF(DABS(DERYM(4)) .LE. ERR) DERYM(4)=0.0D00                       CIFF1960
      IF(DABS(DERYM(5)) .LE. ERR) DERYM(5)=0.0D00                       CIFF1970
      IF(DABS(DERYM(6)*ALPHAC) .LE. ERR) DERYM(6)=0.0D00                CIFF1980
      IF(DABS(DERYM(7)) .LE. ERR) DERYM(7)=0.0D00                       CIFF1990
      IF(DABS(DERYM(8)) .LE. ERR) DERYM(8)=0.0D00                       CIFF2000
      IF(DABS(DERYM(9)*AMBC2) .LE. ERR) DERYM(9)=0.0D00                 CIFF2010
      IF(DABS(DERYM(10)*ALPHAC) .LE. ERR) DERYM(10)=0.0D00              CIFF2020
      IF(DABS(DERYM(11)*AMCO2) .LE. ERR) DERYM(11)=0.0D00               CIFF2030
C                                                                       CIFF2040
C                                                                       CIFF2050
   21 CONTINUE                                                          CIFF2060
C                                                                       CIFF2070
C     ------------------------------------------------------------------CIFF2080
C                                                                       CIFF2090
      NOFNS=NOFNS+1                                                     CIFF2100
C                                                                       CIFF2110
C     ------------------------------------------------------------------CIFF2120
C                                                                       CIFF2130
      RETURN                                                            CIFF2140
      END                                                               CIFF2150
```

```
      SUBROUTINE MPGRM2(N,T,Y,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG, MPGR  10
     1                  JSTART,MAXDER,PW,PPW,LLL,MMM,YF,DELTA,          MPGR  20
     |  DIFFUN,PEDERV,OUTP)                                             MPGR  30
      IMPLICIT REAL*8 (A-H,O-Z)                                         MPGR  40
      EXTERNAL DIFFUN,PEDERV,OUTP                                       MPGR  50
      DIMENSION YF(N)                                                   MPGR  60
      DIMENSION SAVE(12,1),Y(8,1),      YMAX(1),ERROR(1),R(1),          MPGR  70
     1 PW(1),PPW(1),LLL(1),MMM(1)                                       MPGR  80
      COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL                                MPGR  90
      COMMON /BLOK10/ J                                                 MPGR 100
      COMMON NOFNS                                                      MPGR 110
      CALL ERRSET(208,999,-1,1)                                         MPGR 120
      CALL ERRSET(209,999,-1,1)                                         MPGR 130
      FEPS = EPS*1.D+3                                                  MPGR 140
      KFLAG=1                                                           MPGR 150
   80 J = J + 1                                                         MPGR 160
      IF(KFLAG .GT. 0) TO=T                                             MPGR 170
      CALL DIARMF(N,T,Y,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG,JSTART,MPGR 180
     2            MAXDER,PW,PPW,LLL,MMM,DIFFUN,PEDERV)                  MPGR 190
      IF (KFLAG .LT. 0 ) GO TO 4000                                     MPGR 200
      TN = T                                                            MPGR 210
      IF(T .LT. DELTA*0.9999000)GO TO 4000                              MPGR 220
      DT = TN - TO                                                      MPGR 230
      H = DELTA - TO                                                    MPGR 240
      IF(DABS(H) .LT. 1.D-06) GO TO 4000                                MPGR 250
      JSTART = -1                                                       MPGR 260
      DELTA = DELTA + DSEL                                              MPGR 270
      GO TO 80                                                          MPGR 280
 4000 CONTINUE                                                          MPGR 290
      IF(KFLAG.GT.0) GO TO 81                                           MPGR 300
      IF(KFLAG.EQ. -1) JSTART = -1                                      MPGR 310
      IF(KFLAG.EQ. -2) WRITE(6,225)                                     MPGR 320
      IF(KFLAG .EQ. -3)   WRITE(6,230)                                  MPGR 330
        IF(KFLAG.EQ. -4) WRITE(6,235)                                   MPGR 340
        IF(KFLAG.EQ. -4) GO TO 579                                      MPGR 350
```

```
  225 FORMAT(1H-,'THE MAXIMUM ORDER SPECIFIED IS TOO LARGE')
  230 FORMAT(1H-,'CORRECTOR CONVERGENCE NOT ACHIEVED FOR H.GT.HMIN')
  235 FORMAT(1H-,'REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED')
      IF(KFLAG .NE. -1) GO TO 96
      IF(HMIN.LT.1.D-09)  WRITE(6,666)HMIN
  666 FORMAT(1H-,12X,'PROGRAM TERMINATED HMIN = ',D15.6)
      HMIN = HMIN/2.D00
      GO TO 80
   81 CONTINUE
      JSTART = 1
      IF(T .LT. FSEL*0.9999D00) GO TO 95
      DO 222  I = 1,N
  222 YF(I) = Y(1,I)
C-----------------------------------------------------------------------
C THIS SUBROUTINE WRITTEN BY THE USER ALLOWS HIM TO PRINT ANY OF THE
C FOLLOWING DATA:
C        N      = NUMBER OF EQUATIONS
C        T      = INDEPENDENT VARIABLE (NORMALLLY TIME)
C        J      = AMOUNTC OF STEPS TAKEN UP TP THIS POINT IN THE NUM. INT.
C        YF    = A N ELEMENT ARRAY CONTAINING THE DEPENDENT VARIABLES VALUES
C        SAVE   =   ARRAAY (TWO DIMENSIONAL CONTAINING THE DERIVATIVES IN
C                   THE SUBARRAY SAVE(N2,I)) WHERE N2 = N*10 + 1
C        SEL      TIME INTERVAL FOR PRINTING
C        HA     PREVIOUS STEP SIZE
C        NOFNS   NUMBER OF FUNCTION  EVALUATIONS UP TO THIS POINT
      CALL OUTP(N,T,J,YF,SAVE,H,DT,NOFNS)
C-----------------------------------------------------------------------
      FSEL = FSEL + SEL
   95 IF ( T.LT.SPLM) GO TO 80
   96 CONTINUE
 3333 RETURN
  579 EPS = EPS*10.D00
      IF (EPS .GT. FEPS) GO TO 581
      WRITE(6,580)EPS
  580 FORMAT(1X,'NEW TOLERANCE IS' , D15.7)
      JSTART = -1
```

```
      GO TO 80                                                          MPGR 720
  581 WRITE(6,582)FEPS                                                  MPGR 730
  582 FORMAT(1X,'ERROR THAT CAN BE HANDLED GREATER THAN ',D15.7/        MPGR 740
     | 1X,'PROGRAM TERMINATED')                                         MPGR 750
      STOP 3333                                                         MPGR 760
      END                                                               MPGR 770
```

```
************************************
*PROBLEM 5.7    A BIOLOGICAL SYSTEM*
************************************

------------------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     1.000     STEP SIZE = 0.50000D-01     DE. EV. =   62     ACCUM. STEPS =   61      H  = 0.20000D-03
                     PCO2M = 0.451110066333D 02     HCO3PM = 0.247542034915D-01
                     PSCM  = 0.471110066333D 02     HCO3SM = 0.270000000000D-01     HSM = 0.413727319691D-07
                     PCCM  = 0.521110066333D 02     HCO3CM = 0.120000000000D-01     HCM = 0.103543869971D-06
                     PO2M = 0.370169310044D 02     PSOM = 0.320169310044D 02     PCOM = 0.170169310044D 02
------------------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     2.000     STEP SIZE = 0.50000D-01     DE. EV. =  122     ACCUM. STEPS =  121      H  = 0.20000D-03
                     PCO2M = 0.451110066333D 02     HCO3PM = 0.247542034915D-01
                     PSCM  = 0.471110066333D 02     HCO3SM = 0.270000000000D-01     HSM = 0.413727319691D-07
                     PCCM  = 0.521110066333D 02     HCO3CM = 0.120000000000D-01     HCM = 0.103543869971D-06
                     PO2M = 0.370169310044D 02     PSOM = 0.320169310044D 02     PCOM = 0.170169310044D 02
------------------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     3.000     STEP SIZE = 0.50000D-01     DE. EV. =  308     ACCUM. STEPS =  204      H  = 0.35136D-01
                     PCO2M = 0.473383722336D 02     HCO3PM = 0.255123490501D-01
                     PSCM  = 0.484391559105D 02     HCO3SM = 0.270925926239D-01     HSM = 0.421763775951D-07
                     PCCM  = 0.522252314260D 02     HCO3CM = 0.120060803655D-01     HCM = 0.103622415199D-06
                     PO2M = 0.346074423391D 02     PSOM = 0.300425359294D 02     PCOM = 0.163149612503D 02
------------------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     4.000     STEP SIZE = 0.50000D-01     DE. EV. =  404     ACCUM. STEPS =  264      H  = 0.35136D-01
                     PCO2M = 0.477048109455D 02     HCO3PM = 0.256589452597D-01
                     PSCM  = 0.487833228826D 02     HCO3SM = 0.272364101553D-01     HSM = 0.422635912489D-07
                     PCCM  = 0.523740104918D 02     HCO3CM = 0.120189287798D-01     HCM = 0.103786496351D-06
                     PO2M = 0.338817504965D 02     PSOM = 0.292718883367D 02     PCOM = 0.154240532275D 02
```

```
----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     5.000     STEP SIZE = 0.50000D-01     DE. EV. =   470     ACCUM. STEPS =   324      H  = 0.35136D-01
                     PCO2M =  0.478163014308D 02     HCO3PM =  0.257396655861D-01
                     PSCM  =  0.489385942835D 02     HCO3SM =  0.273699469482D-01     HSM =  0.422226631917D-07
                     PCCM  =  0.525231612941D 02     HCO3CM =  0.120327160214D-01     HCM =  0.103961743315D-06
                     PO2M =  0.334646144718D 02     PSOM =  0.287823000721D 02     PCCM =  0.147225262501D 02
----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     6.000     STEP SIZE = 0.50000D-01     DE. EV. =   530     ACCUM. STEPS =   384      H  = 0.35136D-01
                     PCO2M =  0.478910105796D 02     HCO3PM =  0.258082270758D-01
                     PSCM  =  0.490593939301D 02     HCO3SM =  0.274901058121D-01     HSM =  0.421694573853D-07
                     PCCM  =  0.526694187942D 02     HCO3CM =  0.120463997983D-01     HCM =  0.104134921017D-06
                     PO2M =  0.331691245370D 02     PSOM =  0.284257062829D 02     PCCM =  0.141858589689D 02
----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     7.000     STEP SIZE = 0.50000D-01     DE. EV. =   590     ACCUM. STEPS =   444      H  = 0.35136D-01
                     PCO2M =  0.479628645927D 02     HCO3PM =  0.258700437845D-01
                     PSCM  =  0.491733379140D 02     HCO3SM =  0.275983005380D-01     HSM =  0.421256775783D-07
                     PCCM  =  0.528124104085D 02     HCO3CM =  0.120598132874D-01     HCM =  0.104303992740D-06
                     PO2M =  0.329483776203D 02     PSOM =  0.281559734933D 02     PCCM =  0.137714331695D 02
----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     8.000     STEP SIZE = 0.50000D-01     DE. EV. =   650     ACCUM. STEPS =   504      H  = 0.35136D-01
                     PCO2M =  0.480368761197D 02     HCO3PM =  0.259260886235D-01
                     PSCM  =  0.492853590615D 02     HCO3SM =  0.276958974203D-01     HSM =  0.420940033853D-07
                     PCCM  =  0.529522963450D 02     HCO3CM =  0.120729454808D-01     HCM =  0.104468917043D-06
                     PO2M =  0.327804212538D 02     PSOM =  0.279486936597D 02     PCCM =  0.134478486722D 02
```

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

NUMERICAL INTEGRATION SOL.  (GEAR)  :

TIME =      9.000     STEP SIZE = 0.50000D-01     DE. EV. =   710     ACCUM. STEPS =   564       H  = 0.35136D-01
                      PCO2M =  0.481130207140D 02     HCO3PM =  0.259769757604D-01
                      PSCM   =  0.493959602138D 02     HCO3SM =  0.277840697422D-01     HSM =  0.420732705747D-07
                      PCCM   =  0.530892728033D 02     HCO3CM =  0.120858094050D-01     HCM =  0.104629550946D-06
                      PO2M =  0.326514471434D 02     PSOM =  0.277879454416D 02     PCOM =  0.131930343933D 02
-----------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL.  (GEAR)  :

TIME =     10.000     STEP SIZE = 0.50000D-01     DE. EV. =   770     ACCUM. STEPS =   624       H  = 0.35136D-01
                      PCO2M =  0.481905127799D 02     HCO3PM =  0.260232661369D-01
                      PSCM   =  0.495048600045D 02     HCO3SM =  0.278638593528D-01     HSM =  0.420617670704D-07
                      PCCM   =  0.532234847955D 02     HCO3CM =  0.120984172933D-01     HCM =  0.104787329336D-06
                      PO2M =  0.325519245361D 02     PSOM =  0.276626005171D 02     PCOM =  0.129911868369D 02
-----------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL.  (GEAR)  :

TIME =     11.000     STEP SIZE = 0.50000D-01     DE. EV. =   830     ACCUM. STEPS =   684       H  = 0.35136D-01
                      PCO2M =  0.482686153514D 02     HCO3PM =  0.260654716738D-01
                      PSCM   =  0.496117439104D 02     HCO3SM =  0.279362016186D-01     HSM =  0.420579253137D-07
                      PCCM   =  0.533550267396D 02     HCO3CM =  0.121107772441D-01     HCM =  0.104941222561D-06
                      PO2M =  0.324750261175D 02     PSOM =  0.275646214396D 02     PCOM =  0.128307164735D 02
-----------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL.  (GEAR)  :

TIME =     12.000     STEP SIZE = 0.50000D-01     DE. EV. =   890     ACCUM. STEPS =   744       H  = 0.35136D-01
                      PCO2M =  0.483467580576D 02     HCO3PM =  0.261040516214D-01
                      PSCM   =  0.497163911005D 02     HCO3SM =  0.280019350017D-01     HSM =  0.420604248438D-07
                      PCCM   =  0.534839571623D 02     HCO3CM =  0.121228940012D-01     HCM =  0.105091747656D-06
                      PO2M =  0.324157415095D 02     PSOM =  0.274880667620D 02     PCOM =  0.127029426740D 02
```

```
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     13.000    STEP SIZE = 0.50000D-01    DE. EV. =  950    ACCUM. STEPS =  804     H  = 0.35136D-01
                     PCO2M = 0.484245187586D 02    HCO3PM = 0.261394143273D-01
                     PSCM  = 0.498186703550D 02    HCO3SM = 0.280618077866D-01    HSM = 0.420681698522D-07
                     PCCM  = 0.536103118608D 02    HCO3CM = 0.121347701646D-01    HCM = 0.105238985190D-06
                     PC2M = 0.323703198085D 02    PSOM = 0.274284590216D 02    PCCM = 0.126012465332D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     14.000    STEP SIZE = 0.50000D-01    DE. EV. = 1010    ACCUM. STEPS =  864     H  = 0.35136D-01
                     PCO2M = 0.485015876396D 02    HCO3PM = 0.261719212134D-01
                     PSCM  = 0.499185160226D 02    HCO3SM = 0.281164849810D-01    HSM = 0.420802490653D-07
                     PCCM  = 0.537341133420D 02    HCO3CM = 0.121464071361D-01    HCM = 0.105382992637D-06
                     PC2M = 0.323359042626D 02    PSOM = 0.273823678855D 02    PCCM = 0.125205046361D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     15.000    STEP SIZE = 0.50000D-01    DE. EV. = 1070    ACCUM. STEPS =  924     H  = 0.35136D-01
                     PCO2M = 0.485777370994D 02    HCO3PM = 0.262018913035D-01
                     PSCM  = 0.500159070016D 02    HCO3SM = 0.281665555983D-01    HSM = 0.420959003574D-07
                     PCCM  = 0.538553771413D 02    HCO3CM = 0.121578057603D-01    HCM = 0.105523813685D-06
                     PC2M = 0.323102865847D 02    PSOM = 0.273471278786D 02    PCCM = 0.124567003569D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (GEAR) :

TIME =     16.000    STEP SIZE = 0.50000D-01    DE. EV. = 1130    ACCUM. STEPS =  984     H  = 0.35136D-01
                     PCO2M = 0.486527997118D 02    HCO3PM = 0.262296056993D-01
                     PSCM  = 0.501108516420D 02    HCO3SM = 0.282125400065D-01    HSM = 0.421144828289D-07
                     PCCM  = 0.539741159181D 02    HCO3CM = 0.121689667325D-01    HCM = 0.105661484437D-06
                     PC2M = 0.322917379439D 02    PSOM = 0.273206427942D 02    PCCM = 0.124066506929D 02
```

APPENDIX B

This appendix contains:

1) Subroutine DIFSUB. Because this subroutine as it stands can solve systems of up to ten equations, to solve the eleven equations of the "Mathematical model of human muscle-chemical aspects" problem, it was properly modified. Such a modification (only in DIMENSION instructions) is not presented.

2) Subroutine BISYEX. This subroutine drives the Bulirsch and Stoer and polynomial extrapolation methods of DIFSUB, and also implements an algorithm to provide the solution at pre-specified values of the independent variable.

3) As an example, the solution of the problem mentioned in Part 1 is presented. (Refer to Appendix A for a complete list of all the subroutines used by this problem.)

```
      SUBROUTINE DIFSUB (N,T,Y,DY,H,HMIN,EPS,MF,YMAX,ERROR,KFLAG,       DIFS  10
     1                   JSTART,MAXORD,MAXPTS)                          DIFS  20
      IMPLICIT REAL*8 (A-H,O-Z)                                         DIFS  30
C***********************************************************************DIFS  40
C*    THE PARAMETERS TO THIS INTEGRATION SUBROUTINE HAVE               *DIFS  50
C*   THE FOLLOWING MEANINGS                                            *DIFS  60
C*     N        THE NUMBER OF FIRST ORDER FIFFERENTIAL EQUATIONS       *DIFS  70
C*     T         THE INDEPENDENT VARIABLE                              *DIFS  80
C*     Y         THE DEPENDENT VARIABLES, UP TO 10 ARE ALLOWED         *DIFS  90
C*     DY        AN ARRAY OF 10 LOCATIONS WHICH WILL CONTAIN THE       *DIFS 100
C*                 VALUES OF THE DERIVATIVES ON EXIT.                  *DIFS 110
C*     H         THE STEPSIZE THAT SHOULD BE ATTEMPTED. IT MAY BE      *DIFS 120
C*                 INCREASED OR DECREASED BY THE SUBROUTINE.           *DIFS 130
C*     HMIN      THE MINIMUM STEP SIZE THAT SHOULD BE ALLOWE ON THIS   *DIFS 140
C*                 STEP.                                               *DIFS 150
C*     EPS       THE ERROR TEST CONSTANT. THE ESTIMATED ERRORS ARE     *DIFS 160
C*                 REQUIRED TO BE LESS THAN EPS*YMAX IN EACH COMPONENT. *DIFS 170
C*                 THE ERROR TEST WILL BE RELATIVE FOR THOSE COMPONENTS *DIFS 180
C*                 GREATER THAN ONE AND ABSOLUTE FOR THE OTHERS.       *DIFS 190
C*     MF        THE METHOD INDICATOR. THE FOLLOWING ARE ALLOWED..     *DIFS 200
C*                          0   BULIRSCH-STOER RATIONAL EXTRAPOLATION  *DIFS 210
C*                          1   POLYNOMIAL EXTRAPOLATION               *DIFS 220
C*     YMAX      THE MAXIMUM VALUES OF THE DEPENDENT VARIABLES ARE     *DIFS 230
C*                 SAVED IN THIS ARRAY. IT SHOULD BE SET TO + 1 BEFORE *DIFS 240
C*                 THE FIRST ENTRY. (SEE THE DESCRIPTION OF EPS).      *DIFS 250
C*     ERROR     THEESTIMATED SINGLE STEP ERROR IN EACH COMPONENT      *DIFS 260
C*     KFLAG     A COMPLETION CODE WITH  THE FOLLOWING MEANINGS..      *DIFS 270
C*                         -1    THE STEP WAS TAKEN WITH H = HMIN      *DIFS 280
C*                               BUT THE REQUESTED ERROR WAS NOT ACHIEVED.*DIFS 290
C*                         +1    THE STEP WAS SUCCESFUL.               *DIFS 300
      CONTINUE                                                          DIFS 310
C*     JSTART    AN INITIALIZATION INDICATOR WITH THE MEANING..        *DIFS 320
C*                         -1    REPEAT THE LAST STEP, RESTORING THE   *DIFS 330
C*                               VALUES OF Y AND YMAX THAT WERE USED   *DIFS 340
C*                               LAST TIME.                            *DIFS 350
```

```
C*                    +1   TAKE A NEW STEP.                              *DIFS 360
C*      MAXORD THE MAXIMIM ORDER OF EXTRAPOLATION ALLOWED. IT MUST       *DIFS 370
C*                 BE LESS THAN 11.                                      *DIFS 380
C*      MAXPTS THE MAXIMIM NUMBER OF DIFFERENT SUB STEPS SIZES USD       *DIFS 390
C*                 IN THE EXTRAPOLATION PROCESS.                         *DIFS 400
C***********************************************************************DIFS 410
      DIMENSION Y(10),DY(10),YMAX(10),YSAVE(10),YNM1(10),YN(10),DYN(10),DIFS 420
     1          YMAXSV(10),QUOT(11,2),EXTRAP(10,11),YNM1HV(10,12)        DIFS 430
     2         ,YNHV(10,12),YMAXHV(10,12),ERROR(10)                      DIFS 440
C***********************************************************************DIFS 450
C*      THE ARRAYS ARE USED FOR THE FOLLOWING DATA..                     *DIFS 460
C*       YSAVE   THE INITIAL VALUES OF Y ARE SAVED FOR A RESTART         *DIFS 470
C*       YNM1    Y(N-1), THE PREVIOUS VALUE OF Y IN THE MIDPOINT METHOD  *DIFS 480
C*      YN       Y(N),  THE CURRENT VALUE OF Y IN THE MIDPOINT INTEGRAT.*DIFS 490
C*      DYN     THE INITIAL VALUE OF THE DERIVATIVE OF Y.                *DIFS 500
C*      YMAXSV  THE SAVED VALUES OF YMAX AT THE INITIAL POINT.           *DIFS 510
C*      QUOT    THE QUOTIENTS  (H(I)/H(I +M))**2 USED IN THE             *DIFS 520
C*                   EXTRAPOLATION.                                      *DIFS 530
C*      EXTRAP  THE MOST RECENT EXTRAPOLATED VALUES OF Y IN THE CASE     *DIFS 540
C*                 OF POLYNIMIAL EXTRAPOLATION, OR OF THE DIFFERENCES IN *DIFS 550
C*                 THE CASE OF RATIONAL FUNCTION EXTRAPOLATION.          *DIFS 560
C*      YNM1HV  THE VALUES OF YNM1 AT THE MIDPOINT OF THE BASIC INTERVAL*DIFS 570
C*                 IF THE NUMBER OF SUBSTEPS IS DIVISIBLE BY 4. THIS     *DIFS 580
C*                  INFORMATION IS USED TO AVOID REDOING THE INTEGRATION *DIFS 590
C*                  IN CASE THE STEP IS HALVED.                          *DIFS 600
C*      YNHV    THE SIMILAR VALUES OF YN                                 *DIFS 610
C*      YMAXHV  AND THE SAME FOR YMAX                                    *DIFS 620
C*      ERROR   THE ESTIMATES OF THE SINGLE STEP ERRORS ARE SAVED HERE. *DIFS 630
C***********************************************************************DIFS 640
      DATA QUOT/1.,2.25,4.,9.,16.,36.,64.,144.,256.,576.,1024.,          DIFS 650
     1     1.,1.77777777777777,4.,7.111111111111111,                     DIFS 660
     2     16.,28.44444444444444,64.,113.777777777777,                   DIFS 670
     3        256.,455.111111111111,1024./                               DIFS 680
      DATA FMAX/10000000./                                               DIFS 690
C***********************************************************************DIFS 700
C*      FMAX IS A NUMBER SMALLER THAN THE FIRST INTEGER THAT CANNOT BE  *DIFS 710
```

```
C*     REPRESENTED EXACTLY IN FLOATING POINT.                          *DIFS 720
C**********************************************************************DIFS 730
       IF(JSTART .LT. 0) GO TO 2                                        DIFS 740
C**********************************************************************DIFS 750
C*     SAVE THE VALUES OF Y AND YMAX IN CASE A RESTART IS NECESSARY.   *DIFS 760
C**********************************************************************DIFS 770
       DO 1 I = 1,N                                                     DIFS 780
       YSAVE(I) = Y(I)                                                  DIFS 790
     1 YMAXSV(I) = YMAX(I)                                              DIFS 800
       CALL DIFFUN(T,Y,DYN)                                             DIFS 810
       GO TO 4                                                          DIFS 820
C**********************************************************************DIFS 830
C*     RESTORE THE VALUES OF Y AND YMAX FOR A RESTART.                 *DIFS 840
C**********************************************************************DIFS 850
     2 DO 3 I= 1,N                                                      DIFS 860
       Y(I) = YSAVE(I)                                                  DIFS 870
     3 YMAX(I) = YMAXSV(I)                                              DIFS 880
     4 CONTINUE                                                         DIFS 890
C**********************************************************************DIFS 900
C*     THE FOLLOWING COUNTERS AND SWITCHES ARE USED..                  *DIFS 910
C*     J       IS THE COUNT THROUGH THE DIFFERENT SUB STEPS G USED.    *DIFS 920
C*     JODD    IS 1 IF J IS ODD, 2 IF J IS EVEN                        *DIFS 930
C*     JHVSV   IS THE NUMBER OF SUBSTEP SIZES FOR WHICH HALF WAY       *DIFS 940
C*               INFORMATION HAS BEEN SAVED.                           *DIFS 950
C*     JHVSV1  THE VALUE OF JHVSV FROM THE PEEVIOUS CYCLE              *DIFS 960
C*     M       THE NUMBER OF PAIRS OS SUBSTEPS WHICH MAKE UP THE STEP H*DIFS 970
C*               M TAKES THE SEQUENCE 1,2,3,4,6,8,12,16,  ETC.         *DIFS 980
C*     MNEXT   THE NEXT VALUE OF M                                     *DIFS 990
C*     MTWO    THE NEXT BUT ONE VALUE OF M.                            *DIFS1000
C*     QUOTSV  THE LAST VALUE OF GUCUT IS IRREGULAR DUE TO THE FACT    *DIFS1010
C*               THAT THE SEQUENCE BY THE MULTIPLES 9/4,16/9 (/ODD) OR *DIFS1020
C*               16/9, 9/4 (EVEN UNTIL THE FINAL MULTIPLE OF 4. HOWEVER*DIFS1030
C*               (H(0)/H(M))**2 IS ALWAYS M**2. THE REGULAR VALUE OF   *DIFS1040
C*               QUOT IS SAVED IN QUOTSV, AND REPLACED BY M**2.        *DIFS1050
C*     KONV    IS SET TO +1 INITIALLY, AND RESET TO -1 IF THE  ERROR   *DIFS1060
C*               TEST FAILS                                            *DIFS1070
```

```
C***********************************************************************CIFS1080
    5 JHVSV1 = 0                                                        CIFS1090
      KFLAG = 1                                                         CIFS1100
    6 JHVSV = 0                                                         CIFS1110
    7 A = H + T                                                         CIFS1120
      JODD = 1                                                          CIFS1130
      M = 1                                                             CIFS1140
      MNEXT = 2                                                         CIFS1150
      MTWO = 3                                                          CIFS1160
      DO 23 J = 1,MAXPTS                                                CIFS1170
      QUOTSV = QUOT(J,JODD)                                             CIFS1180
      QUOT(J,JODD) = M*M                                                CIFS1190
      KONV=1                                                            CIFS1200
      IF( J.LE.(MAXCRD/2)) KONV = -1                                    CIFS1210
      IF ( J.LE.(MAXCRD+1)) GO TO 8                                     CIFS1220
      L = MAXCRD + 1                                                    CIFS1230
      HCHNGE = .70710678D0*HCHNGE                                       CIFS1240
      GO TO 9                                                           CIFS1250
    8 L = J                                                             CIFS1260
      HCHNGE = 1.0D0 + (MAXCRD + 1 - J)/6.0D0                           CIFS1270
    9 B = H/M                                                           CIFS1280
      C = B*0.5D0                                                       CIFS1290
      IF (J.GT.JHVSV1) GO TO 11                                         CIFS1300
C***********************************************************************CIFS1310
C*     THE VALUES OF THE MIDPOINT INTEGRATION WERE SAVED AT THE        *CIFS1320
C*     HALF WAY POINT IN THE PREVIOUS INTEGRATION. USE THEM.           *CIFS1330
C***********************************************************************CIFS1340
      DO 10 I=1,N                                                       CIFS1350
        YN(I) = YNHV(I,J)                                               CIFS1360
      YNM1(I) = YNM1HV(I,J)                                             CIFS1370
   10  YMAX(I) = YMAXHV(I,J)                                            CIFS1380
      GO TO 16                                                          CIFS1390
C***********************************************************************CIFS1400
C*     INTEGRATE OVER THE RANGE H BY 2*M STEPR OF A MIDPOINT METHOD.   *CIFS1410
C***********************************************************************CIFS1420
   11 DO 12 I= 1,N                                                      CIFS1430
```

```
      YNM1(I) = YSAVE(I)                                                CIFS1440
      YN(I) = YSAVE(I) + G*DYN(I)                                       CIFS1450
   12 YMAX(I) = YMAXSV(I)                                               CIFS1460
      M2 = M + M                                                        CIFS1470
      TU = T                                                            CIFS1480
      DO 15 K = 2,M2                                                    CIFS1490
      TU = TU + G                                                       CIFS1500
      CALL DIFFUN(TU,YN,DY)                                             CIFS1510
      DO 13 I = 1,N                                                     CIFS1520
      U = YNM1(I) + B*DY(I)                                             CIFS1530
      YNM1(I) = YN(I)                                                   CIFS1540
      YN(I) = U                                                         CIFS1550
      U = DABS(U)                                                       CIFS1560
   13 IF(U.GT.YMAX(I)) YMAX(I) = U                                      CIFS1570
      IF((K.NE.M).OR.(JHVSV1.NE.0).OR.(K.EQ.3)) GO TO 15                CIFS1580
      JHVSV = JHVSV + 1                                                 CIFS1590
      DO 14 I = 1,N                                                     CIFS1600
      YNHV(I,JHVSV) = YN(I)                                             CIFS1610
      YNM1HV(I,JHVSV) = YNM1(I)                                         CIFS1620
   14  YMAXHV(I,JHVSV) = YMAX(I)                                        CIFS1630
   15 CONTINUE                                                          CIFS1640
   16 CALL DIFFUN(A,YN,DY)                                              CIFS1650
      DO 22 I = 1,N                                                     CIFS1660
      V = EXTRAP(I,1)                                                   CIFS1670
C***********************************************************************CIFS1680
C*      CALCULATE THE FINAL VALUE TO BE USED IN THE EXTRAPOLATION PROC. *CIFS1690
C***********************************************************************CIFS1700
      TA = (YN(I) + YNM1(I) + G*DY(I))*0.5D0                            CIFS1710
      C = TA                                                            CIFS1720
C***********************************************************************CIFS1730
C*      INSERT THE INTEGRAL AS THE FIRST EXTRAPOLATED VALUE.            *CIFS1740
C***********************************************************************CIFS1750
      EXTRAP(I,1) = TA                                                  CIFS1760
      IF (L.LT.2) GO TO 21                                              CIFS1770
      IF (DABS(V)*FMAX.LT.DABS(C)) GO TO 27                             CIFS1780
      IF (MF.GT.0) GO TO 19                                             CIFS1790
```

```
C**********************************************************************DIFS1800
C*      PERFORM THE EXTRAPOLATION BY RATIONAL FUNCTIONS ON THE         *DIFS1810
C*       SECOND AND SUBSEQUENT INTEGRALS.                              *DIFS1820
C**********************************************************************DIFS1830
         DO 18 K = 2,L                                                 DIFS1840
         B1 = QUOT(K,JODD)*V                                           DIFS1850
         B = B1 -C                                                     DIFS1860
         U = V                                                         DIFS1870
         IF (B.EQ.0) GO TO 17                                          DIFS1880
         B = (C - V)/B                                                 DIFS1890
         U = C*B                                                       DIFS1900
         C = B1*B                                                      DIFS1910
      17 V = EXTRAP(I,K)                                               DIFS1920
         EXTRAP(I,K) = U                                               DIFS1930
         TA = TA + U                                                   DIFS1940
      18 CONTINUE                                                      DIFS1950
         GO TO 21                                                      DIFS1960
C**********************************************************************DIFS1970
C*      PERFORM THE EXTRAPOLATION BY POLLYNOMIALS ON THE               *DIFS1980
C*       SECOND AND SUBSEQUENT INTEGRALS.                              *DIFS1990
C**********************************************************************DIFS2000
      19 DO 20 K = 2,L                                                 DIFS2010
         TA = TA + (TA-V)/(QUOT(K,JODD) - 1.D0)                        DIFS2020
         V = EXTRAP(I,K)                                               DIFS2030
         EXTRAP(I,K) = TA                                              DIFS2040
      20 CONTINUE                                                      DIFS2050
         GO TO 21                                                      DIFS2060
      21 U = DABS(TA)                                                  DIFS2070
         IF(U.GT.YMAX(I)) YMAX(I) = U                                  DIFS2080
         ERROR(I)  = DABS(Y(I) - TA)                                   DIFS2090
         Y(I) = TA                                                     DIFS2100
         IF (ERROR(I).GT.EPS*YMAX(I)) KONV=-1                          DIFS2110
      22 CONTINUE                                                      DIFS2120
         QUOT(J,JODD) = QUOTSV                                         DIFS2130
         IF (KONV.GT.0) GO TO 25                                       DIFS2140
         JODD = 3 - JODD                                               DIFS2150
```

```
      M = MNEXT                                                         DIFS2160
      MNEXT = MTWO                                                      DIFS2170
      MTWO = M + M                                                      DIFS2180
   23 CONTINUE                                                          DIFS2190
      JHVSV1 = JHVSV                                                    DIFS2200
   24 IF (DABS(H).LE.HMIN) GO TO 26                                     DIFS2210
      H = H*0.5D0                                                       DIFS2220
      IF (DABS(H).GE.HMIN) GO TO 6                                      DIFS2230
      H = DSIGN(HMIN,H)                                                 DIFS2240
      GO TO 5                                                           DIFS2250
   25 H = H*HCHNGE                                                      DIFS2260
      T = A                                                             DIFS2270
      RETURN                                                            DIFS2280
   26 KFLAG = -1                                                        DIFS2290
      GO TO 25                                                          DIFS2300
   27 QUOT(J,JCOD) = QUOTSV                                             DIFS2310
      GO TO 24                                                          DIFS2320
      END                                                               DIFS2330
```

```
      SUBROUTINE BISYEX (N,T,Y,DY,H,HMIN,EPS,MF,YMAX,ERROR,KFLAG,       BISY  10
     1             JSTART,MAXORD,MAXPTS,DELTA)                          BISY  20
      IMPLICIT REAL*8 (A-H,O-Z)                                         BISY  30
      DIMENSION Y(1),DY(1) ,YF(5)                                       BISY  40
      DIMENSION YMAX(1)                                                 BISY  50
      DIMENSION ERROR(1)                                                BISY  60
      COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL                                BISY  70
      COMMON /BLCK10/ J                                                 BISY  80
      COMMON NOFNS                                                      BISY  90
      CALL ERRSET(208,999,-1,1)                                         BISY 100
      CALL ERRSET(209,999,-1,1)                                         BISY 110
      JSTART = 1                                                        BISY 120
      KFLAG=1                                                           BISY 130
   80 J = J + 1                                                         BISY 140
      IF(KFLAG .GT. 0) TO=T                                             BISY 150
      CALL GDFSUB(N,T,Y,DY,H,HMIN,EPS,MF,YMAX,ERROR,KFLAG,              BISY 160
     1             JSTART,MAXORD,MAXPTS)                                BISY 170
  701 IF (KFLAG .LT.0 ) GO TO 4000                                      BISY 180
  702 TN = T                                                            BISY 190
      IF(T .LT. DELTA*0.9999D00)GO TO 81                                BISY 200
  703 DT = TN - TO                                                      BISY 210
      H = DELTA - TO                                                    BISY 220
      IF(DABS(H) .LT. 1.D-06) GO TO 81                                  BISY 230
  705 JSTART = -1                                                       BISY 240
      T = TO                                                            BISY 250
      DELTA = DELTA + DSEL                                              BISY 260
      GO TO 80                                                          BISY 270
 4000 CONTINUE                                                          BISY 280
      JSTART = -1                                                       BISY 290
      IF(HMIN.LT.1.D-10) GO TO 2593                                     BISY 300
      H = H/1.0D+1                                                      BISY 310
      HMIN = HMIN/1.D+01                                                BISY 320
      GO TO 80                                                          BISY 330
   81 CONTINUE                                                          BISY 340
      JSTART = 1                                                        BISY 350
```

```
      IF(T .LT. FSEL*0.99990D0) GO TO 95                                BISY 360
      DO 222  I = 1,N                                                   BISY 370
  222 YF(I) = Y(  I)                                                    BISY 380
      CALL OUTP (N,T,J,YF,DY,H,DT,NOFNS)                                BISY 390
      FSEL = FSEL + SEL                                                 BISY 400
   95 IF ( T.LT.SPLM) GO TO 80                                          BISY 410
   96 CONTINUE                                                          BISY 420
      GO TO 2599                                                        BISY 430
 2593 WRITE(6,2597)                                                     BISY 440
 2597 FORMAT(2H-,5X,'INTEGRATION REQUIRES TOO SMALL STEP SIZE,',1X,     BISY 450
     | 'PROGRAM TERMINATED')                                            BISY 460
 2599 CONTINUE                                                          BISY 470
 9000 RETURN                                                            BISY 480
      END                                                               BISY 490
```

```
      C     ------------------------------------------------------------
      C     ------------------------------------------------------------
      C
            IMPLICIT REAL*8(A-H,O-Z)
            REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMO2,MTMCO2,KCSHM
            REAL*4 PRIRA
      C
            COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAC,HB,APN2,ATN2
            COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
            COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
            COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MC,PSOM,HCO3SM,HSM,
           1PCOM,HCO3CM,HCM,PSOM,PSOMO,PCOM,PCOMO,PN2M,PN2MC,PSNM,PSNMC,PCNM,
           2PCNMC
            COMMON/BLOCK5/YM(11)
            COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSCM,DSCCM,
           1   CM,VMB,CMMAX,CPSNM,CSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,CSPBM,
           2   DSPHM,DCSBM,KCSHM,FM
            COMMON/BLOCK8/FFTIM
            COMMON /BLOCK9/ FSEL,SPLM,CSEL,SEL
            COMMON/BLOK10/J
            COMMON/BLOK11/T
            COMMON/BLOK12/DELT
            COMMON NCFNS
      C
      C
      C     ------------------------------------------------------------
      C
      C
            WRITE(6,610)
        610 FORMAT(1H1,/1X,36('*')/
           |                 1X,'*PROBLEM 3.7     A BIOLOGICAL SYSTEM*'/
           | 1X,36('*')//)
            CALL DATA
      C
            TIME=0.0000
            FTIME=16.0000
            FFTIM=2.05000
            DELT=0.05000
            T=0.0000
            SPLM=0.0000
            SEL = 1.0000
            PRIRA = SEL/DELT
            KCS = 0
      C
      C
      C     ------------------------------------------------------------
      C
      C 5   IF(SPLM .EQ. 0.0000) CALL INTAL
            SPLM=SPLM+DELT
            IF(SPLM .GT. FTIME) GO TO 40
            IF(DABS(SPLM-FFTIM) .LE. 0.0001D00) GO TO 2
            GO TO 1
      C
        2   CONTINUE
```

```
0034                CALL FORCE
            C
0035        1       CONTINUE
0036                IF(KOS .LT. IFIX(4.000*PRIRA))GO TO 11
0037                KOS = 0
0038                WRITE(6,215)
0039          215   FORMAT(1H1///)
0040           11   CONTINUE
0041                KOS = KOS + 1
            C
            C     ---------------------------------------------------------------------
            C
0042                CALL MUSCLE
            C
            C     ---------------------------------------------------------------------
            C
0043                GO TO 5
            C
0044          40    CONTINUE
0045                STOP
0046                END
```

```
0001            SUBROUTINE MUSCLE
          C
          C     ------------------------------------------------------------------
          C
          C
0002            IMPLICIT REAL*8(A-H,O-Z)
0003            REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMC2,MTMCC2,KCSHM,MTMMO2
0004            REAL*8 INTAMA,INTAMM,MM,MMS,MMC
0005            REAL*8 L1,L2
0006            REAL*4 PW,PPW
          C
          C     ------------------------------------------------------------------
          C
0007            COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2
0008            COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
0009            COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
0010            COMMON/BLOCK4/PCC2M,HCO3PM,CARBM,PO2M,CO2M,PC2MC,PSCM,HCC3SM,HSM,
               1PCCM,HCC3CM,HCM,PSCM,PSOMC,PCCM,PCCMC,PN2M,PN2MC,PSNM,PSNMC,PCNM,
               2PCNMC
0011            COMMON/BLOCK5/YM(11)
0012            COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,DSCCM,
               1   QM,VMB,CMMAX,DPSNM,DSCNM,MTMC2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,
               2   DSPHM,DCSBM,KCSHM,FM
0013            COMMON/BLOCK8/FFTIM
0014            COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL
0015            COMMON/BLOK10/J
0016            COMMON/BLCK11/T
0017            COMMON/BLCK12/DELT
0018            COMMON NOFNS
          C
0019            DIMENSION       DY(11),YF(11),YMAX(11),ERROR(11)
          C
          C
          C     ------------------------------------------------------------------
          C
          C
0020            IF(T    .EQ. 0.0000) GO TO 111
0021            GO TO 222
0022       111  NOFNS=0
0023            J=0
0024            JSTART=1
          C
          C
0025            FSEL=SEL
0026            N=11
          C
          C
0027            MF = 0
0028            MAXDRD = 6
0029            MAXPTS = 5
          C
0030            RMIN=1.0D-10
0031            EPS = 1.D-05
0032            MAXDER=6
```

```
0033                HMAX=DELT
0034                DSEL=DELT
0035                H=1.0D-04
0036                DELTA = DSEL
0037                DO 10 I=1,11
0038                YMAX(I)=1.0D00
0039         10     CONTINUE
0040         222    CONTINUE
             C
0041                IF(T .GT. 0.0D00) H=DELT
0042                IF(DABS(T-FFTIM) .LE. DELT) H=1.0D-04
             C
             C     ----------------------------------------------------------------
             C
0043                CALL       BISYEX (N,T,YM,DY,H,HMIN,EPS,MF,YMAX,ERROR,KFLAG,
                  1            JSTART,MAXORD,MAXPTS,DELTA)
             C
             C     ----------------------------------------------------------------
             C
             C     ----------------------------------------------------------------
             C
0044         100    CONTINUE
             C
0045                RETURN
0046                END
```

```
0001                SUBROUTINE OUTP(N,T,J,YF,DY,  H,DT,NCFNS)
              C
              C     ------------------------------------------------------------
              C
0002                IMPLICIT REAL*8(A-H,O-Z)
              C
0003                COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSOM,HCO3SM,HSM,
                   1PCOM,HCO3CM,HCM,PSOM,PSOMC,PCOM,PCOMC,PN2M,PN2MC,PSNM,PSNMO,PCNM,
                   2PCNMC
              C
0004                DIMENSION YF(1),DY(11)
              C
              C     ------------------------------------------------------------
              C
              C
              C
0005                WRITE(6,8)
0006          8     FORMAT(1X,125('-')/)
              C
0007                WRITE(6,5)
0008          5     FORMAT(1X,'NUMERICAL INTEGRATION SOL.   (EXTRAPOLATION)    :')
0009                WRITE(6,1)T,DT,NCFNS,J,H
0010          1     FORMAT(//1X,'TIME =',F10.3,5X,'STEP SIZE =',D12.5,5X,'DE. EV. =',
                   1          I6,5X,'ACCUM. STEPS =',I6,5X,' H  =',D12.5/)
              C
0011                WRITE(6,2)(YF(I),I=1,8)
0012          2     FORMAT(22X,'PCO2M =',D20.12,5X,'HCO3PM =',D20.12//
                   1       22X,'PSOM  =',D20.12,5X,'HCO3SM =',D20.12,5X,'HSM =',D20.12
                   2       //22X,'PCOM  =',D20.12,5X,'HCO3CM =',D20.12,5X,'HCM =',
                   3       D20.12/)
              C
              C
0013                WRITE(6,22)(YF(I),I=9,11)
0014          22    FORMAT(22X,'PO2M =',D20.12,5X,'PSOM =',D20.12, 5X,'PCOM =',D20.12)
              C
              C
              C
              C     ------------------------------------------------------------
              C
0015                RETURN
0016                END
```

```
*************************************
*PROBLEM 3.7    A BIOLOGICAL SYSTEM*
*************************************

-----------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     1.000     STEP SIZE = 0.50000D-01     DE. EV. =  1093     ACCUM. STEPS =    53      H  = 0.75000D-01
                     PCO2M =  0.451110066333D 02     HCO3PM =  0.247542034915D-01
                     PSCM  =  0.471110066333D 02     HCO3SM =  0.270000000000D-01     HSM =  0.413727319691D-07
                     PCCM  =  0.521110066333D 02     HCO3CM =  0.120000000000D-01     HCM =  0.103543869971D-06
                     PO2M =  0.370169310044D 02     PSOM =  0.320169310044D 02     PCOM =  0.170169310044D 02
-----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     2.000     STEP SIZE = 0.50000D-01     DE. EV. =  1913     ACCUM. STEPS =    93      H  = 0.75000D-01
                     PCO2M =  0.451110066333D 02     HCO3PM =  0.247542034915D-01
                     PSCM  =  0.471110066333D 02     HCO3SM =  0.270000000000D-01     HSM =  0.413727319691D-07
                     PCCM  =  0.521110066333D 02     HCO3CM =  0.120000000000D-01     HCM =  0.103543869971D-06
                     PO2M =  0.370169310044D 02     PSOM =  0.320169310044D 02     PCOM =  0.170169310044D 02
-----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     3.000     STEP SIZE = 0.50000D-01     DE. EV. =  3789     ACCUM. STEPS =   166      H  = 0.66667D-01
                     PCO2M =  0.473387153011D 02     HCO3PM =  0.255121510796D-01
                     PSCM  =  0.484396161042D 02     HCO3SM =  0.270925791103D-01     HSM =  0.421769700529D-07
                     PCCM  =  0.522254051006D 02     HCO3CM =  0.120060949487D-01     HCM =  0.103622607553D-06
                     PO2M =  0.346077887271D 02     PSOM =  0.300427186787D 02     PCOM =  0.163145313747D 02
-----------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     4.000     STEP SIZE = 0.50000D-01     DE. EV. =  5033     ACCUM. STEPS =   208      H  = 0.66667D-01
                     PCO2M =  0.477049274867D 02     HCO3PM =  0.256588776759D-01
                     PSCM  =  0.487834776819D 02     HCO3SM =  0.272363736203D-01     HSM =  0.422638013151D-07
```

```
------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     5.000     STEP SIZE = 0.50000D-01     DE. EV. =  6249     ACCUM. STEPS =   251      H  = 0.75000D-01
                     PCO2M =  0.478164020811D 02     HCO3PM =  0.257396419838D-01
                     PSCM  =  0.489387044714D 02     HCO3SM =  0.273699114626D-01     HSM =  0.422227958943D-07
                     PCCM  =  0.525233398514D 02     HCO3CM =  0.120327323701D-01     HCM =  0.103961961463D-06
                     PO2M =  0.334646139899D 02     PSOM =  0.287823013170D 02     PCOM =  0.147224389547D 02
------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     6.000     STEP SIZE = 0.50000D-01     DE. EV. =  7493     ACCUM. STEPS =   293      H  = 0.66667D-01
                     PCO2M =  0.478911062169D 02     HCO3PM =  0.258082146418D-01
                     PSCM  =  0.490594952923D 02     HCO3SM =  0.274900771979D-01     HSM =  0.421695678948D-07
                     PCCM  =  0.526695905868D 02     HCO3CM =  0.120464154646D-01     HCM =  0.104135131370D-06
                     PO2M =  0.331691157401D 02     PSOM =  0.284257563468D 02     PCOM =  0.141858139761D 02
------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     7.000     STEP SIZE = 0.50000D-01     DE. EV. =  8733     ACCUM. STEPS =   336      H  = 0.75000D-01
                     PCO2M =  0.479629579349D 02     HCO3PM =  0.258700360532D-01
                     PSCM  =  0.491734373706D 02     HCO3SM =  0.275982788289D-01     HSM =  0.421257775324D-07
                     PCCM  =  0.528125755394D 02     HCO3CM =  0.120598282631D-01     HCM =  0.104304195043D-06
                     PO2M =  0.329483762673D 02     PSOM =  0.281559708871D 02     PCOM =  0.137714109247D 02
------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     8.000     STEP SIZE = 0.50000D-01     DE. EV. =  9953     ACCUM. STEPS =   378      H  = 0.66667D-01
                     PCO2M =  0.480369677773D 02     HCO3PM =  0.259260841382D-01
                     PSCM  =  0.492854576464D 02     HCO3SM =  0.276958817093D-01     HSM =  0.420940956632D-07
                     PCCM  =  0.529524552938D 02     HCO3CM =  0.120729598148D-01     HCM =  0.104469111817D-06
                     PO2M =  0.327804252001D 02     PSOM =  0.279487350086D 02     PCOM =  0.134478399861D 02
```

```
----------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =      9.000     STEP SIZE = 0.50000D-01     DE. EV. = 11197     ACCUM. STEPS =   420      H  = 0.66667D-01
                      PCO2M =  0.481131106249D 02     HCO3PM =  0.259769739032D-01
                      PSCM  =  0.493960577886D 02     HCO3SM =  0.277840591124D-01     HSM =  0.420733562477D-07
                      PCCM  =  0.530894260449D 02     HCO3CM =  0.120858231495D-01     HCM =  0.104630138771D-06
                      PO2M =  0.326514540056D 02     PSOM =  0.277880583868D 02     PCOM =  0.131930343620D 02
----------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     10.000     STEP SIZE = 0.50000D-01     DE. EV. = 12413     ACCUM. STEPS =   463      H  = 0.75000D-01
                      PCO2M =  0.481906007979D 02     HCO3PM =  0.260232664690D-01
                      PSCM  =  0.495049563098D 02     HCO3SM =  0.278638530085D-01     HSM =  0.420618468651D-07
                      PCCM  =  0.532236327424D 02     HCO3CM =  0.120984304940D-01     HCM =  0.104787510723D-06
                      PO2M =  0.325519361176D 02     PSOM =  0.276626328899D 02     PCOM =  0.129911931283D 02
----------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     11.000     STEP SIZE = 0.50000D-01     DE. EV. = 13657     ACCUM. STEPS =   505      H  = 0.66667D-01
                      PCO2M =  0.482687013991D 02     HCO3PM =  0.260654738348D-01
                      PSCM  =  0.496118387452D 02     HCO3SM =  0.279361988837D-01     HSM =  0.420579998323D-07
                      PCCM  =  0.533551697447D 02     HCO3CM =  0.121107899403D-01     HCM =  0.104941397948D-06
                      PO2M =  0.324750388153D 02     PSOM =  0.275646904617D 02     PCOM =  0.128307266380D 02
----------------------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL.  (EXTRAPOLATION)   :

TIME =     12.000     STEP SIZE = 0.37500D-01     DE. EV. = 14921     ACCUM. STEPS =   548      H  = 0.37500D-01
                      PCO2M =  0.483468420899D 02     HCO3PM =  0.261040553069D-01
                      PSCM  =  0.497164843219D 02     HCO3SM =  0.280019353019D-01     HSM =  0.420604946658D-07
                      PCCM  =  0.534840955298D 02     HCO3CM =  0.121229062267D-01     HCM =  0.105091917417D-06
                      PO2M =  0.324157565575D 02     PSOM =  0.274880821386D 02     PCOM =  0.127029557520D 02
```

```
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     13.000      STEP SIZE = 0.50000D-01      DE. EV. = 16117     ACCUM. STEPS =    590       H  = 0.66667D-01
                       PCO2M =  0.484246007727D 02      HCO3PM =  0.261394192784D-01
                       PSCM  =  0.498187618642D 02      HCO3SM =  0.280618106331D-01     HSM =  0.420682354460D-07
                       PCCM  =  0.536104458555D 02      HCO3CM =  0.121347819486D-01     HCM =  0.105239149653D-06
                       PO2M =  0.323703349381D 02     PSOM =  0.274285029131D 02     PCOM =  0.126012612680D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     14.000      STEP SIZE = 0.50000D-01      DE. EV. = 17361     ACCUM. STEPS =    632       H  = 0.66667D-01
                       PCO2M =  0.485016676631D 02      HCO3PM =  0.261719272081D-01
                       PSCM  =  0.499186057683D 02      HCO3SM =  0.281164899563D-01     HSM =  0.420803108729D-07
                       PCCM  =  0.537342431988D 02      HCO3CM =  0.121464185046D-01     HCM =  0.105383152088D-06
                       PO2M =  0.323359188183D 02     PSOM =  0.273824551758D 02     PCOM =  0.125205203265D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     15.000      STEP SIZE = 0.50000D-01      DE. EV. = 18601     ACCUM. STEPS =    675       H  = 0.75000D-01
                       PCO2M =  0.485778151376D 02      HCO3PM =  0.262018981695D-01
                       PSCM  =  0.500159949297D 02      HCO3SM =  0.281665623557D-01     HSM =  0.420959587218D-07
                       PCCM  =  0.538555030661D 02      HCO3CM =  0.121578167362D-01     HCM =  0.105523968378D-06
                       PO2M =  0.323103027107D 02     PSOM =  0.273471479588D 02     PCOM =  0.124567169836D 02
------------------------------------------------------------------------------------------------------------
NUMERICAL INTEGRATION SOL. (EXTRAPOLATION)   :

TIME =     16.000      STEP SIZE = 0.50000D-01      DE. EV. = 19821     ACCUM. STEPS =    717       H  = 0.66667D-01
                       PCO2M =  0.486528758269D 02      HCO3PM =  0.262296132529D-01
                       PSCM  =  0.501109377465D 02      HCO3SM =  0.282125482301D-01     HSM =  0.421145381379D-07
                       PCCM  =  0.539742381005D 02      HCO3CM =  0.121689773364D-01     HCM =  0.105661634602D-06
                       PO2M =  0.322917533581D 02     PSOM =  0.273206940540D 02     PCOM =  0.124066675388D 02
```

APPENDIX C

This appendix contains:

1) Listing of DESUB system. Only the double precision version is presented.

2) Subroutine MPDDSB to drive DESUB.

3) As an example of this system, the solution to "A stirred tank reactor with exothermic reaction" problem is presented.

The subroutine SEDDS used by this program corresponds to a modified version of DDEOUT from DESUB for a better printer spacing handling.

```
      SUBROUTINE DDE(XFX,N,Y,XI,XF,HI,EPS,DERR,XOUTX)                   DDE   10
C                                                                       DDE   20
C     DOUBLE PRECISION VERSION OF DE                                    DDE   30
C                                                                       DDE   40
C     USING DDE CAUSES VALUES OF DEPENDENT VARIABLES TO BE PRINTED AT   DDE   50
C     POINTS AUTOMATICALLY SELECTED BY THE PROGRAM.                     DDE   60
C                                                                       DDE   70
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ DDE   80
      IMPLICIT REAL*8 (A-H,O-Z)                                         DDE   90
      DIMENSION Y(11)                                                   DDE  100
      DOUBLE PRECISION XI,XF,HI,EPS,Y                                   DDE  110
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ DDE  120
      EXTERNAL XFX,DERR,XOUTX                                           DDE  130
      DOUBLE PRECISION SP                                               DDE  140
      COMMON /DDESPC/ NP, KOUNT                                         DDE  150
      KOUNT = 0                                                         DDE  160
      NP = 0                                                            DDE  170
      SP = 0.0D0                                                        DDE  180
      CALL XDDE(SP,XFX,N,Y,XI,XF,HI,EPS,DERR,XOUTX)                     DDE  190
      RETURN                                                            DDE  200
      END                                                               DDE  210
```

```
      SUBROUTINE DDESP(SP,XFX,N,Y,XI,XF,HI,EPS,DERR,XOUTX)            DDES  10
      IMPLICIT REAL*8 (A-H,O-Z)                                       DDES  20
C                                                                     DDES  30
C     DOUBLE PRECISION VERSION OF DESP                                DDES  40
C                                                                     DDES  50
C     USING DDESP CAUSES VALUES OF DEPENDENT VARIABLES TO BE PRINTED AT DDES  60
C     SPECIFIED POINTS IN ADDITION TO THOSE AUTOMATICALLY SELECTED.   DDES  70
C                                                                     DDES  80
      DIMENSION Y(11)                                                 DDES  90
      DOUBLE PRECISION Y,HI,EPS                                       DDES 100
      EXTERNAL XFX,DERR,XOUTX                                         DDES 110
      DOUBLE PRECISION SP,XF,XI                                       DDES 120
      COMMON /DDESPC/ NP, KOUNT                                       DDES 130
      NP = 1                                                          DDES 140
      KOUNT = 0                                                       DDES 150
       IF(SP*(XF-XI)) 2,4,10                                          DDES 160
    2 SP = DSIGN(SP,XF-XI)                                            DDES 170
      GO TO 10                                                        DDES 180
    4 IF(SP .NE. 0.0D0) GOTO 10                                       DDES 190
      NP = 0                                                          DDES 200
   10 CALL XDDE(SP,XFX,N,Y,XI,XF,HI,EPS,ERR,XOUTX)                    DDES 210
      RETURN                                                          DDES 220
      END                                                             DDES 230
```

```
      SUBROUTINE XODE(SP,XFX,N,Y,XI,XF,HI,EPS,DERR,XOUTX)                 XODE  10
      IMPLICIT REAL*8 (A-H,O-Z)                                           XODE  20
C                                                                         XODE  30
C     DOES DOUBLE PRECISION HOUSKEEPING DURING INTEGRATION,               XODE  40
C     CHECKS INPUT PARAMETERS, CHECKS FOR SPECIFIED OUTPUT                XODE  50
C     POINTS, STEPS H, ETC.                                               XODE  60
C                                                                         XODE  70
      DIMENSION Y(11),DY(11),S(11),R(11),YR(11)                           XODE  80
      DOUBLE PRECISION Y,DY,YR,XI,XF,XIPRT,XFPRT,TTL,X,XP,XPMX,XR,XT,S,   XODE  90
     1R,DZCT,DP2,DEMIN,DEMAX,EX,ER,EH,CHDIV,DZCTUP,HI,HIPRT,HMIN,HMAX,HPXODE 100
     2,HQ,HR,FH,EPS,EPSPRT,SP,H,SPPRT                                     XODE 110
      COMMON /DDESPC/ NP, KOUNT                                           XODE 120
      COMMON /IPARAM/M,NMAX                                               XODE 130
      COMMON /DPARAM/DZCT,DP2,DEMAX,DEMIN,CHDIV,DZCTUP                    XODE 140
      COMMON /DINFO/  EX,ER,EH,NE,NERR                                    XODE 150
      COMMON /OUTPUT/ SPPRT,HIPRT,XIPRT,XFPRT,EPSPRT,NPPRT,TITLE          XODE 160
      LOGICAL STYPE,KONVF,TITLE                                           XODE 170
      EXTERNAL XFX,DERR                                                   XODE 180
C*********INITIALIZE CONSTANTS ****************************************** XODE 190
      DZCT = 2.77D-17                                                     XODE 200
      DP2 = 32768.DO                                                      XODE 210
      DEMAX = 1.DO-02                                                     XODE 220
      DEMIN = 1.DO-18                                                     XODE 230
      CHDIV = 1024.DOO                                                    XODE 240
      DZCTUP = 3.6D16                                                     XODE 250
      M = 6                                                               XODE 260
      NMAX = 20                                                           XODE 270
C******** CHECK PARAMETERS AND PERFORM INITIALIZATION          ********** XODE 280
      TITLE=.TRUE.                                                        XODE 290
      STYPE=.TRUE.                                                        XODE 300
      NPPRT=NP                                                            XODE 310
      SPPRT=SP                                                            XODE 320
      HIPRT=HI                                                            XODE 330
      XIPRT=XI                                                            XODE 340
      XFPRT=XF                                                            XODE 350
```

```
      EPSPRT=EPS
      IF ((N.LE.0).OR.(N.GT.NMAX)) GOTO 84
      IF ((EPS.LT.DEMIN).OR.(EPS.GT.DEMAX)) GOTO 85
      TTL=XF-XI
      H=HI
      IF (TTL*H) 86,87,12
   12 IF (((H/TTL)* DP2 .LT.1.).OR.((H/TTL).GT.1.)) GOTO 88
      DO 14 I=1,N
      S(I)=DABS(Y(I))
   14 CONTINUE
      KONVF=.TRUE.
      HMIN=H/DHDIV
      HMAX=TTL
      HP=0.
      XP=XI
      X=XI
C******** BEGIN SOLUTION OF DIFFERENTIAL EQUATIONS          ************
   20 H=H/3.
      H=3.*H
      IF ((NP.EQ.0).AND.(.NOT.STYPE)) GOTO 50
      XPMX=XP-X
      FH=XPMX/H
      IF (FH.GT.DZOT) GOTO 50
C******** GET TO SPECIFIED VALUE OF X AND CALL XOUTX        *************
   30 IF (DABS(FH).GT.DZOT) GOTO 34
      DO 32 I=1,N
      YR(I)=Y(I)
   32 CONTINUE
      HQ=HP
      XR=X
      GOTO 36
   34 HQ=XPMX+HP
      HR=HQ
      XR=XT
      CALL DREDIF (N,XR,YR,DY,HR,HQ,EPS,M,S,R,KONVF,XFX,DERR)
      HQ=XR-XT
```

```
   36   CALL XFX (YR,XR,CY)                                                  XODE 720
        STYPE=.TRUE.                                                         XODE 730
        CALL XOUTX(YR,DY,N,XR,STYPE)                                         XODE 740
        STYPE=.FALSE.                                                        XODE 750
        IF (KCNVF) GOTO 40                                                   XODE 760
   38   IF (KCNVF) GOTO 70                                                   XODE 770
        GOTO 82                                                              XODE 780
C******** STEP TO NEXT SPECIFIED OUTPUT VALUE                      ********* XODE 790
   40   IF ((XF-XR)/TTL.LE.0.D0) GOTO 70                                     XODE 800
        KOUNT = KOUNT+1                                                      XODE 810
        XP = XI+FLOAT(KOUNT)*SP                                              XODE 820
        IF ((XP-XF)/H.GT.0.D0) XP=XF                                         XODE 830
        GOTO 20                                                              XODE 840
C******** OUTPUT RESULTS AT A NATURAL STEP                           ******* XODE 850
   50   IF ((DABS((X-XR)/H)).LE.DZOT) GOTO 60                                XODE 860
        CALL XFX (Y,X,DY)                                                    XODE 870
        CALL XOUTX(Y,DY,N,X,STYPE)                                           XODE 880
C******** PERFORM A STEP IN SOLUTION OF DIFFERENTIAL EQUATIONS       ******* XODE 890
   60   IF ((XF-X)/TTL.LE.0.D0) GOTO 70                                      XODE 900
        IF (DABS(H).LT.HMIN) H=DSIGN(HMIN,H)                                 XODE 910
        IF (DABS(H).GT.HMAX) H=DSIGN(HMAX,H)                                 XODE 920
        IF ((XF-X-H)/TTL.LT.0.D0) H=XF-X                                     XODE 930
        XT=X                                                                 XODE 940
        CALL ODESUB (N,X,Y,DY,H,HMIN,EPS,M,S,R,KCNVF,XFX,DERR)               XODE 950
        HP=X-XT                                                              XODE 960
        IF (KCNVF) GOTO 20                                                   XODE 970
        GOTO 80                                                              XODE 980
C******** STANDARD RETURN ON COMPLETION OF SOLUTION                ******** XODE 990
   70   RETURN                                                               XODE1000
C******** SET UP ERROR RETURN AND DIAGNOSTICS                      ******** XODE1010
   80   NERR=1                                                               XODE1020
        ER=0.D0                                                              XODE1030
         DO 81 I=1,N                                                         XODE1040
        IF (ER*S(I).GE.R(I)) GOTO 81                                         XODE1050
        ER=R(I)/S(I)                                                         XODE1060
        NE=I                                                                 XODE1070
```

```
 81   CONTINUE                                   XDCE1080
      EH=HP                                      XDCE1090
      EX=X                                       XDCE1100
      CALL  XFX (Y,X,DY)                         XDCE1110
      CALL XOUTX(Y,DY,N,X,STYPE)                 XDCE1120
      GOTO 92                                    XDCE1130
 82   NERR=1                                     XDCE1140
      ER=0.00                                    XDCE1150
      DO 83 I=1,N                                XDCE1160
      IF (ER*S(I).GE.R(I)) GOTO 83               XDCE1170
      ER=R(I)/S(I)                               XDCE1180
      NE=I                                       XDCE1190
 83   CONTINUE                                   XDCE1200
      EH=HQ                                      XDCE1210
      EX=XR                                      XDCE1220
      GOTO 92                                    XDCE1230
 84   NERR=2                                     XDCE1240
      GOTO 90                                    XDCE1250
 85   NERR=3                                     XDCE1260
      GOTO 90                                    XDCE1270
 86   NERR=4                                     XDCE1280
      GOTO 90                                    XDCE1290
 87   NERR=5                                     XDCE1300
      GOTO 90                                    XDCE1310
 88   NERR=6                                     XDCE1320
 90   CALL XFX (Y,XI,DY)                         XDCE1330
      CALL XOUTX(Y,DY,N,XI,STYPE)                XDCE1340
 92   CALL DERROR                                XDCE1350
      RETURN                                     XDCE1360
      END                                        XDCE1370
```

```
      SUBROUTINE DDESUB(N,X,Y,DY,H,HMIN,EPS,JM,S,R,KONVF,FCT,DERR)      DDES  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         DDES  20
C                                                                       DDES  30
C**************************PERFORM INITIALIZAION TO CALL DDERSB*********DDES  40
C                                                                       DDES  50
      COMMON/DDERCM/ YA(11),SA(11),DZ(11),JMAX                          DDES  60
      DIMENSION Y(11),DY(11),R(11),S(11)                                DDES  70
      DOUBLE PRECISION Y,S,YA,SA,DZ                                     DDES  80
      EXTERNAL FCT,DERR                                                 DDES  90
C***********************************************************************DDES 100
C***********FOR AN EXTRAPOLATION OF ORDER JM, JM+1 UNEXTRAPOLATED*******DDES 110
C           APPROXIMATIONS ARE REQUIRED. THREE MORE ARE ALLOWED IN      DDES 120
C           ATTEMPTING TO ACHIEVE CONVERGENCE                           DDES 130
      JMAX = JM + 4                                                     DDES 140
C**********SAVE THE INITIAL VALUES FOR THE DEPENDENT VARIABLES AND******DDES 150
C           THE ERROR TEST VESTOR FOR THE STEP                          DDES 160
      DO 100 I = 1,N                                                    DDES 170
      YA(I) = Y(I)                                                      DDES 180
      SA(I) = S(I)                                                      DDES 190
100   CONTINUE                                                          DDES 200
C***********USE THE FUNCTION ROUTINE TO OBTAIN THE INITIAL SLOPES*******DDES 210
C     DZ = DY/DX                                                        DDES 220
      CALL FCT(Y,X,DZ)                                                  DDES 230
C**********PERFORM AN INTEGRATION STEP *******************************<DDES 240
      CALL DDERSB(N,X,Y,DY,H,HMIN,EPS,JM,S,R,KONVF,FCT,DERR)            DDES 250
      RETURN                                                            DDES 260
      END                                                               DDES 270
```

```
      SUBROUTINE DREDIF(N,X,Y,DY,H,HMIN,EPS,JM,S,R,KONVF,FCT,DERR)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C************PERFORM INITIALIZATION TO CALL DDERSB ********************
      COMMON/DDERCM/ YA(11),SA(11),DZ(11),JMAX
      DIMENSION Y(11),DY(11),R(11),S(11)
      DOUBLE PRECISION Y,YA,SA,DZ
      EXTERNAL FCT,DERR
      DO 300 I = 1,N
      Y(I) = YA(I)
  300 CONTINUE
C**********PERFORM AN INTEGRATION STEP ********************************
      CALL DDERSB(N,X,Y,DY,H,HMIN,EPS,JM,S,R,KONVF,FCT,DERR)
      RETURN
      END
```

```
      SUBROUTINE DDERSB(N,X,Y,DY,H,HMIN,EPS,JM,S,R,KONVF,FCT,DERR)       DDER  10
C                                                                        DDER  20
C           IMPLEMENTS THE ALGORITHM IN DOUBLE PRECISION                 DDER  30
C           AND PERFORM ONE INTEGRATION STEP                             DDER  40
C                                                                        DDER  50
      COMMON/DDERCM/ YA(11),SA(11),DZ(11),JMAX                           DDER  60
      COMMON / DPARAM / DZOT,DP2,DEMAX,DEMIN,DHDIV,DZCTUP                DDER  70
      DIMENSION Y(11),DY(11),S(11),R(11),        YL(11),YM(11),          DDER  80
     |          D(7),DT(11,7),YG(11,8),YH(11,8),SG(11,8)                 DDER  90
      DOUBLE PRECISION Y,DY,S,R,YA,YL,YM,DZ,SA,D,DT,YG,YH,SG,X,H,HMIN,A,DDER 100
     1EPS,FC,G,B,B1,XU,U,V,C,TA,DZOT,DP2,DEMIN,DEMAX,DHDIV,DZCTUP        DDER 110
      LOGICAL KONVF,KONV,BO,BH                                           DDER 120
C***************THE LOGICAL VARIABLE, BH, DETERMINES WHETER THE ********DDER 130
C           STEPSIZE HAS BEEN HALVED. INITIALLY FALSE.                   DDER 140
C           LATER BH IS FALSE IF THE STEPSIZE IS CUT BY A FACTOR NOT 2   DDER 150
 10   BH = .FALSE.                                                       DDER 160
C********** PRESET THE CONVERGENCE SUCCESS FLAG TRUE *******************DDER 170
      KONVF = .TRUE.                                                     DDER 180
C**********ADVANCE THE INDEPENDENT VARIABLE BY THE STEPSIZE, H *********DDER 190
 20   A=H + X                                                            DDER 200
C**********SET THE SWITCH BO FOR THE FIRST SET OF COEFFICIENTS,D********DDER 210
      BO = .FALSE.                                                       DDER 220
C**********INITIALIZE THE H SEQUENCE   H/M,H/JR,H/JS *******************DDER 230
      M = 1                                                              DDER 240
      JR = 2                                                             DDER 250
      JS = 3                                                             DDER 260
C**********JJ IS THE INDEX FOR THE ARRAY OF VALUES SAVED IN*************DDER 270
C           CASE THE INTERVAL MUST BE HALVED                             DDER 280
      JJ = 0                                                             DDER 290
C**********************************************************************DDER 300
C                              INTEGRATION STEP                          DDER 310
C                          MIDPOINT + EXTRAPOLATION                      DDER 320
C**********************************************************************DDER 330
      DO 200 J = 1,JMAX                                                  DDER 340
C********************************************************************* DDER 350
```

```
C*********** SET THE VALUES OF THE EXTRAPOLATION COEFFICINTS TO *******CDER 360
C           THEIR CORRECT VALUES FOR THIS EXTRAPOLATION STEP            CDER 370
      IF(.NOT. BO)GO TO 201                                             CDER 380
      D(2) = 16.D0/9.D0                                                 CDER 390
      D(4) = 64.D0/9.D0                                                 CDER 400
      D(6) = 256.D0/9.D0                                                CDER 410
      GOTO 202                                                          CDER 420
  201 D(2) = 9.D0/4.D0                                                  CDER 430
      D(4) = 9.D0                                                       CDER 440
      D(6) = 36.D0                                                      CDER 450
C*********** IF THE ORDER OF THE EXTRAPOLATION STEP BEING COMPLTED IS **CDER 460
C            LESS THAN JM/2, SET KONV FALSE                             CDER 470
  202 KONV = .TRUE.                                                     CDER 480
      IF(J.LE.(JM/2)) KONV = .FALSE.                                    CDER 490
      IF(J.LE.(JM+1)) GOTO 203                                          CDER 500
C**********************************************************************CDER 510
C        RESTRICT THE ORDER OF THE EXTRAPOLATION TO JM *****************CDER 520
C                 ADJUST THE EXTRAPOLATION COEFFFICIENT                 CDER 530
        L = JM + 1                                                      CDER 540
      D(L) = 4.D0*D(L-2)                                                CDER 550
C**********DISCOURAGE THE STEP-INCREASING FACTOR,FC, BY A FACTOR OF ****CDER 560
C           SQRT(2) SINCE CONVERGENCE WAS NOT OBTAINED I  JM            CDER 570
C            EXTRAPOLATIONS                                             CDER 580
      FC= .7071068D0*FC                                                 CDER 590
      GOTO 204                                                          CDER 600
C***********THE NUMBER, J, OF EXTRAPOLATIONS HAS NOT EXCEEDED JM ******CDER 610
C           FIND D(J) = (H DIVIDED BY H/M)**2                           CDER 620
C           ADJUST THE FACTOR, FC,USED TO ADJUST THE STEPSIZE FOR       CDER 630
C           THE NEXT STEP TO BE TAKEN                                   CDER 640
  203 L = J                                                             CDER 650
        D(L) = FLOAT(M*M)                                               CDER 660
      FC = 1.D0 + FLOAT(JM + 1 - J)/6.D0                                CDER 670
C**********************************************************************CDER 680
C           MODIFIED MIDPOINT RULE USED FO  FIND FIRST                  CDER 690
C           VALUE FOR THE THIS EXTRAPOLATION STEP                       CDER 700
C**********************************************************************CDER 710
```

```
      204    M = M+M                                                        DDER 720
             G = H/FLOAT(M)                                                 DDER 730
             B = G + G                                                      DDER 740
      C********** IF THE STEPSIZE HAS NOT BEEN HALVED OR IF THE ORDER OF THE *DDER 750
      C             EXTRAPOLATION STEP EXEEEDS THAT FOR WHICH PREVIOUSLY    DDER 760
      C               COMPUTED VALUES WERE SAVED, THEY MUST BE COMPUTED     DDER 770
             IF ((.NOT.BH).OR.(J.GE.(JMAX-1))) GOTO 205                     DDER 780
      C********OTHERWISE THE VALUES HAVE BEEN SAVED AND CAN BE RESTORED ******DDER 790
             DO 210 I = 1,N                                                 DDER 800
             YM(I) = YH(I,J)                                                DDER 810
             YL(I) = YG(I,J)                                                DDER 820
             S(I) = SG(I,J)                                                 DDER 830
      210    CONTINUE                                                       DDER 840
             GOTO 206                                                       DDER 850
      C********COMPUTE SAARTING VALUES FOR THE MODIFIED MIDPOINT RULE ********DDER 860
      205    DO 220 I = 1,N                                                 DDER 870
             YL(I) = YA(I)                                                  DDER 880
             YM(I) = YA(I) + G*DZ(I)                                        DDER 890
             S(I) = SA(I)                                                   DDER 900
      220    CONTINUE                                                       DDER 910
             KH = M/2                                                       DDER 920
             XU = X                                                         DDER 930
      C*********THE MEMBER OF THE H SEQUENCE BEING USED BY THE MIDPOINT ******DDER 940
      C             INTEGRATION RULE IS H/M. COMPUTE TO THE END OF THE STEP DDER 950
      C             CONTINUOUSLY UPDATING THE VESTOR,S, CONTAINING THE LARGEST-  DDER 960
      C             VALUE-SO-FAR FOR EACH DEPENDENT VARIABLE                DDER 970
             DO 230 K = 2,M                                                 DDER 980
             XU = XU + G                                                    DDER 990
             CALL FCT(YM,XU,DY)                                             DDER1000
             DO 231 I = 1,N                                                 DDER1010
             U = YL(I) + B*DY(I)                                            DDER1020
             YL(I) = YM(I)                                                  DDER1030
             YM(I) = U                                                      DDER1040
             U = DABS(U)                                                    DDER1050
             IF(U.GT.S(I)) S(I) = U                                         DDER1060
      231    CONTINUE                                                       DDER1070
```

```
C*********IN CASE THE INTERVAL MUST BE HALVED NEXT TIME. SAVE THE *******CDER1080
C          VALUES AT HALFWAY ALONG(KH*M/2) THE STEP UNLESS K = 3          CDER1090
      IF ((K.NE.KH).OR.(K.EQ.3)) GOTO 230                                 CDER1100
      JJ = 1 + JJ                                                         CDER1110
      DO 232 I = 1,N                                                      CDER1120
      YH(I,JJ) = YM(I)                                                    CDER1130
      YG(I,JJ) = YL(I)                                                    CDER1140
      SG(I,JJ) = S(I)                                                     CDER1150
232   CONTINUE                                                            CDER1160
230   CONTINUE                                                            CDER1170
206   CALL FCT(YM,A,DY)                                                   CDER1180
      DO 240 I = 1,N                                                      CDER1190
C*********IS USED TO SAVE THE VALUE OBTAINED BY THE MIDPOINT RULE *******CDER1200
C          USING THE PREVIOUS MEMBER OF THE H SEQUENCE                    CDER1210
C          THE FIRST TIME THROUGH THIS VALUE IS MEANINGLESS, BUT IT       CDER1220
C          IS NOT USED SINCE L IS LESS THAN 2)                            CDER1230
          V = DT(I,1)                                                     CDER1240
C*********COMPUTE THE FINAL VALUE OBTAINED FOR THIS MEMBER OF THE *******CDER1250
C          H SEQUENCE BY THE MODIFIED MIDPOINT RULE'                      CDER1260
      DT (I,1) = (YM(I) + YL(I) + G*DY(I)) *.5                            CDER1270
      C = DT(I,1)                                                         CDER1280
      TA = C                                                              CDER1290
C********* AT LEAST TWO VALUES ARE NEEDED TO START EXTRAPOLATION ********CDER1300
      IF (L.LT. 2) GOTO 242                                               CDER1310
C**********IF THE VALUE JUST COMPUTED BY THE MIDPOINT RULE SHOWS A *****CDER1320
C          LARGE JUMP FROM HE PREVIOUS, HALVE THE INTERVAL                CDER1330
      IF((DABS(V)*DZCTUP.LT.DABS(C)).AND.(H.NE.HMIN).AND.(J.GT.JM/2+1)) CDER1340
     1             GO TO 30                                               CDER1350
C**********PERFORM THE L STEPS FOR THE CURRENT LTH ORDER ***************CDER1360
C          EXTRAPOLATION STEP. IF THE DENOMINATOR OF THE RATIONAL         CDER1370
C          FUNCTION GOES TO ZERO AT ANY STEP, SET DT AT THAT STEP         CDER1380
C          TO ITS VALUE JUST BEFORE                                       CDER1390
      DO 241 K = 2,L                                                      CDER1400
      B1 = D(K)*V                                                         CDER1410
      B = B1 -C                                                           CDER1420
      U = V                                                               CDER1430
```

```
      IF(B.EQ.0.D0) GO TO 243                                           CDER1440
      B = (C-V)/B                                                       CDER1450
      U = C*B                                                           CDER1460
      C = B1*B                                                          CDER1470
243   V = DT(I,K)                                                       CDER1480
      DT(I,K) = U                                                       CDER1490
      TA = U + TA                                                       CDER1500
241   CONTINUE                                                          CDER1510
C**********USE THE ERROR ROUTINE FOR EACH DEPENDENT VARIABLE TO CHECK***CDER1520
C         WHETER CONVERGENCE HAS BEEN ACHIEVED                          CDER1530
 242  CALL DERR(TA,Y(I),S(I),R(I),EPS,KONV)                             CDER1540
240   CONTINUE                                                          CDER1550
      IF(KONV) GO TO 40                                                 CDER1560
C********RESET THE EXTRAPOLATION COEFFICIENTS **************************CDER1570
      D(3) = 4.D0                                                       CDER1580
      D(5) = 16.D0                                                      CDER1590
C********FLIP THE BO SWITCH FOR THR NEXT SET OF COEFFICIENTS ***********CDER1600
      BO = (.NOT.BO)                                                    CDER1610
C********TAKE THE NEXT MEMBER OF THE H SEQUENCE ************************CDER1620
      M = JR                                                            CDER1630
      JR = JS                                                           CDER1640
      JS = M + M                                                        CDER1650
C*********AND GO BACK FOR THE NEXT EXTRAPOLATION ***********************CDER1660
200   CONTINUE                                                          CDER1670
C********IF, AFTER ALL THE EXTRAPOLATIONS ALLOWED, CONVERGENCE HAS *****CDER1680
C        NOT BEEN ACHIEVED, ATTEMPT TO HALVE H SO THAT THE SAVED        CDER1690
C        VALUES CAN BE USED (SET BH TRUE FOR THIS PURPOSE)              CDER1700
C        IF HALVING H MAKES IT LESS THAN HMIN, SET H = HMIN             CDER1710
C        IN THIS CASE THE SAVED VALUES CANNOT BE USED                   CDER1720
C        IF H HAD ALREADY BEEN AT HMIN, CONVERGENCE CANNOT BE           CDER1730
C        ACHIEVED FOR THIS HMIN AND THIS EPS CRITERION. SET KONV FALSE  CDER1740
      BH = (.NOT.BH)                                                    CDER1750
30    IF(DABS(H).LE.HMIN) GO TO 50                                      CDER1760
      H = H/2.D0                                                        CDER1770
      IF (DABS(H).GE.HMIN) GO TO 20                                     CDER1780
      H = DSIGN(HMIN,H)                                                 CDER1790
```

```
      GO TO 1C                                                          CDER18CC
50    KONVF = .FALSE.                                                   CDER1810
C********WHETER CR NCT CCNVERGENCE HAS BEEN ACHIEVED ******************CDER182C
C         SET A NEW SUGGESTEC STEPSIZE FCR THE NEXT STEP                CDER183C
C         ASSIGN THE END OF STEPVALUE TO THE INCEPENDENT VARIALBLE      CDER1840
40    H= FC*H                                                           CDER185C
      X = A                                                             CDER186C
      RETURN                                                            CDER1870
      END                                                               CDER188C
```

```
      SUBROUTINE DDEOUT(Y,DY,N,X,STYPE)                                 DDEO  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         DDEO  20
C                                                                       DDEO  30
C        OUTPUT ROUTINE FOR DOUBLE PRECISION RESULTS                    DDEO  40
C                                                                       DDEO  50
      DIMENSION Y(11),DY(11)                                            DDEO  60
      DOUBLE PRECISION Y,DY,X,SP,H,XI,XF,EPS                            DDEO  70
      LOGICAL STYPE,TITLE                                               DDEO  80
      COMMON / OUTPUT/ SP,H,XI,XF,EPS,NP,TITLE                          DDEO  90
C******** IF TITLE = .TRUE. WRITE HEADINGS AND SET TITLE = .FALSE. *****DDEO 100
      IF(.NOT. TITLE) GO TO 10                                          DDEO 110
      TITLE = .FALSE.                                                   DDEO 120
      WRITE(6,89) N,XI,XF,EPS,H                                         DDEO 130
      IF(NP.EQ.1) WRITE(6,88) SP                                        DDEO 140
      WRITE(6,87)                                                       DDEO 150
C********DETERMINES WHETER OUTPUT POINT IS SPECIFIED POINT**************DDEO 160
C        OR NATURAL STEP AND OUTPUT IT                                  DDEO 170
   10 IF((NP.EQ.1).AND.STYPE)GO TO 20                                   DDEO 180
      WRITE(6,86) X,(Y(I),I=1,N)                                        DDEO 190
      RETURN                                                            DDEO 200
   20 WRITE(6,85)X,(Y(I),I =1,N)                                        DDEO 210
      RETURN                                                            DDEO 220
   89 FORMAT(1H1,10X,19HDE SOLUTION FOR N =,I2,24H EQUATIONS FROM XSTARTDDEO 230
     1 =,D12.5,10H TO XEND =,D12.5/8X,31HWITH LOCAL ERROR TOLERANCE EP =DDEO 240
     2,1PD12.5,26H AND INITIAL STEP SIZE H =,0PD12.5,1H./8X,44HPRINTING DDEO 250
     3OCCURS AT EACH NATURAL STEP IN TIME)                              DDEO 260
   88  FORMAT (1H+,52X,37HAND AT SPECIFIED POINTS (XSART+K*SP)/8X,22HFORDDEO 270
     1 K=0,1,... AND SP =,D12.5,42H (SPECIFIED POINTS ARE IDENTIFIED WITDDEO 280
     2H *).)                                                            DDEO 290
   87 FORMAT(1H0,14X,47HTHE OUTPUT COLUMNS ARE  X, Y(1), Y(2),..., Y(N)/DDEO 300
     1)                                                                 DDEO 310
   86 FORMAT(1H ,10X,4(D25.16,5X)/(41X,3(D25.16,5X)))                   DDEO 320
   85 FORMAT(1H ,4X,1H*,5X,4(D25.16,5X)/(41X,3(D25.16,5X)))             DDEO 330
      END                                                               DDEO 340
```

```
      SUBROUTINE DERROR                                                 DERR  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         DERR  20
C                                                                       DERR  30
C         WRITES AN ERROR DIAGNOSTIC IN THE CASE OF ERROR FAILURE       DERR  40
C         IN DOUBLE PRECISION                                           DERR  50
C                                                                       DERR  60
      DOUBLE PRECISION EX,EH,ER                                         DERR  70
      COMMON/DINFO/EX,ER,EH,NE,NERR                                     DERR  80
      GOTO(10,20,30,40,50,60),NERR                                      DERR  90
10    WRITE(6,91)EX,EH,ER,NE                                            DERR 100
91    FORMAT(5H0****,5X,35HNO CONVERGENCE IN ABOVE STEP TO X =,D12.5, 9HDERR 110
     1 WITH H =,D12.5, 1H,,5X, 4H****,/10X,22HTHE LIMITING ERROR IS ,   DERR 120
     2D12.5,13H IN EQUATION ,I2//)                                      DERR 130
      RETURN                                                            DERR 140
20    WRITE(6,92)                                                       DERR 150
92    FORMAT(5H0****,5X,19HN.LT.0 .OR. N.GT.20,5X,4H****)               DERR 160
      RETURN                                                            DERR 170
30    WRITE(6,93)                                                       DERR 180
93    FORMAT(5H0****,5X,29HEP.LT.1.D-18 .OR. EP.GT.1.D-2,5X,4H****)     DERR 190
      RETURN                                                            DERR 200
40    WRITE(6,94)                                                       DERR 210
94    FORMAT(5H0****,5X,22HH*(XEND-XSTART) .LT. 0,5X,4H****)            DERR 220
      RETURN                                                            DERR 230
50    WRITE(6,95)                                                       DERR 240
95    FORMAT(5H0****,5X,21HH=0. .OR. XEND=XSTART,5X,4H****)             DERR 250
      RETURN                                                            DERR 260
   60 WRITE(6,96)                                                       DERR 270
96    FORMAT(5H0****,5X,48HH.LT.(XEND-XSTART)/2**15 .OR. H.GT.(XEND-XSTADERR 280
     1RT),5X,4H****)                                                    DERR 290
      RETURN                                                            DERR 300
      END                                                               DERR 310
```

```
      SUBROUTINE DSE(TA,Y,S,R,EPS,KONV)                                 DSE   10
      IMPLICIT REAL*8 (A-H,O-Z)                                         DSE   20
C                                                                       DSE   30
C        CHECKS FOR DOUBLE PRECISION CONVERGENCE USING                  DSE   40
C        SATANDARD ERROR TEST                                           DSE   50
C                                                                       DSE   60
      DOUBLE PRECISION TA,S,U                                           DSE   70
      U = DABS(TA)                                                      DSE   80
      IF(U .GT. S) S = U                                                DSE   90
      CALL DERR(TA,Y,S,R,EPS,KONV)                                      DSE  100
      RETURN                                                            DSE  110
      END                                                               DSE  120
```

```
      SUBROUTINE DRE( TA,Y,S,R,EPS,KONV)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C        CHECKS FOR DOUBLE PRECISION CONVERGENCE USING
C        RELATIVE ERROR TEST
C
      DOUBLE PRECISION Y,S
      S = DABS(Y)
      CALL DERR(TA,Y,S,R,EPS,KONV)
      RETURN
      END
```

```
      SUBROUTINE DAE(TA,Y,S,R,EPS,KONV)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C        CHECKS FOR DOUBLE PRECISION CONVERGENCE USING
C        ABSOLUTE ERROR TEST
C
      DOUBLE PRECISION S
      S = 1.0D0
      CALL DERR(TA,Y,S,R,EPS,KONV)
      RETURN
      END
```

```
      SUBROUTINE CERR(TA,Y,S,R,EPS,KONV)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C        PERFORMS DOUBLE PRECISION CONVERGENCE TEST
C
       DOUBLE PRECISION TA,Y,S,R,EPS
      LOGICAL KONV
      R = DABS (Y - TA)
      Y = TA
      IF(S.LT.EPS) S = EPS
      IF(R.GT. EPS*S)KONV = .FALSE.
      RETURN
      END
```

```
      SUBROUTINE   MPCDSB(FEVAL,N,YSTART,XSTART,XEND,H,EPS,DSE,DDECUT)  MPCD  10
      IMPLICIT REAL*8 (A-H,O-Z)                                         MPCD  20
      INTEGER*4 ENCA                                                    MPCD  30
      DOUBLE PRECISION XSTART,XEND,H,EPS,YSTART                         MPCD  40
      DIMENSION YSTART(01)                                              MPCD  50
      DIMENSION ENCA(20)                                                MPCD  60
      COMMON NOFNS                                                      MPCD  70
      EXTERNAL FEVAL,DDECUT,DSE                                         MPCD  80
      CALL ERRSET(208,999,-1,1)                                         MPCD  90
      CALL ERRSET(209,999,-1,1)                                         MPCD 100
 1000 READ(5,3900,END=5000) ( ENCA(I), I= 1,20)                         MPCD 110
      NOFNS = 0                                                         MPCD 120
 3900 FORMAT(20A4)                                                      MPCD 130
      WRITE(6,3950) ENCA                                                MPCD 140
 3950 FORMAT(1H1,20A4)                                                  MPCD 150
      READ(5,4000)N,XSTART,XEND,H,EPS                                   MPCD 160
 4000 FORMAT(I2,4X,4(D12.4,4X))                                         MPCD 170
      READ(5,4010) (YSTART(I), I= 1,N)                                  MPCD 180
 4010 FORMAT( 5(D12.4,4X))                                              MPCD 190
      CALL DDE(FEVAL,N,YSTART,XSTART,XEND,H,EPS,DSE,DDECUT)             MPCD 200
      WRITE(6,97) NOFNS                                                 MPCD 210
 97   FORMAT(1H0,5X,36HTOTAL NO OF FUNCTION EVALUATIONS IS ,I6)         MPCD 220
      GO TO 1000                                                        MPCD 230
 5000 RETURN                                                            MPCD 240
      END                                                               MPCD 250
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION YSTART(2)
      EXTERNAL FEVAL,SECDS ,DSE
      COMMON /PRINT2/ SEL
      SEL = 0.1000
      CALL       MPCDSB(FEVAL,N,YSTART,XSTART,XEND,H,EPS,DSE,SECDS )
      STOP
      END
```

```
      SUBROUTINE FEVAL(Y,X,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(1), DY(1)
      COMMON NOFNS
      NOFNS = NOFNS + 1
      ZE =     DEXP(50.D00*(.5D00 - (1.D00/Y(   2))))
      DY(2)= -2.D00*(Y(   2) - 1.75D00) - 30.D00*(Y(   2) - 1.75D00)*
     1(Y(   2) - 2.D00) + Y(   1)*ZE
      DY(1) = (1.D00 - Y(   1) - Y(   1)*ZE)
      RETURN
      END
```

```
PROBLEM 3.2  A STIRRED TANK REACTOR WITH EXOTHERMIC REACTION.


   DE SOLUTION FOR N = 2 EQUATIONS FROM XSTART = 0.0          TO XEND = 0.10000D 02
  WITH LOCAL ERROR TOLERANCE EP = 1.00000D-05 AND INITIAL STEP SIZE H = 0.10000D-02.
  PRINTING OCCURS AT EACH NATURAL STEP IN TIME

       THE OUTPUT COLUMNS ARE  X, Y(1), Y(2),..., Y(N)

      0.1133300781249999D 00       0.9968891542570408D 00       0.1754524729554151D 01
      0.2574926757812498D 00       0.9928911305563112D 00       0.1766523647817972D 01
      0.3872390136718747D 00       0.9885513103139648D 00       0.1787911077032473D 01
      0.5818585205078121D 00       0.9764666153592469D 00       0.1846290520575783D 01
      0.8737877807617181D 00       0.9052598588310059D 00       0.1961681467445490D 01
      0.1311681671142577D 01       0.5433450606283630D 00       0.2068434421553161D 01
      0.1822557876586912D 01       0.4532981015105606D 00       0.2001930238648728D 01
      0.2335434082031247D 01       0.4909607271921399D 00       0.1955418157856489D 01
      0.2929456321716305D 01       0.5016340594138951D 00       0.1999328461068078D 01
      0.3624815601348873D 01       0.5003157327516672D 00       0.2000148248714282D 01
      0.4436068094253535D 01       0.4999317221019638D 00       0.2000005115976391D 01
      0.5652946833610527D 01       0.5000025178889163D 00       0.1999999649274659D 01
      0.7072638696193685D 01       0.5000004036273776D 00       0.2000000092654519D 01
      0.8965561179637894D 01       0.5000001778329324D 00       0.1999999932902291D 01
      0.1000000000000000D 02       0.5000001983739569D 00       0.2000000014330724D 01

 TOTAL NO OF FUNCTION EVALUATIONS IS     783
```

APPENDIX D

This appendix contains:

1) Runge-Kutta-Merson algorithm (programmed using PL/1).

2) Solution of "The thermal decomposition of ozone" problem.

Note: Refer to IBM Scientific Subroutine Package for Runge-Kutta-Gill algorithm.

```
    INTEG: PROC OPTIONS(MAIN);
 DCL XO BIN(53),HEADING CHAR(100), N FIXED,
     (DELOUT,DELT,ER,FINTIM) BIN(53) EXT;
 DELOUT,DELT,ER,FINTIM=0;
 NEXT: GET DATA(HEADING,N);  PUT EDIT(HEADING)(PAGE,A);
 BEGIN; DCL YO(N) BIN(53), FUN ENTRY;
     GET LIST(XO,YO) COPY;  PUT SKIP;
     GET DATA COPY;  /*DELOUT,DELT,ER,FINTIM*/
     IF FINTIM=0 & DELT=0 & DELOUT=0 THEN DELT=1E-2;
     IF ER=0 THEN ER=1E-3;
     IF DELT=0 & FINTIM ¬=0 THEN DELT=(FINTIM-XO)/1E3;
     IF FINTIM=0 THEN FINTIM=XO+DELT*1E3;
     IF DELOUT=0 THEN DELOUT=DELT*10;
     CALL RUNGE(XO,YO,FUN,N);
 END;  GOTO NEXT;
 END INTEG;
*  PROCESS;
   (NOUFL):
 RUNGE: PROC(XO,YO,FUN,N);
 DCL  (XO,YO(*),(Y,K1,K2,K3,K33,K4,YOLD,Y3)(N),H,DELSUM,EA,IR) BIN(53),
     (IT,IIT,IIIT) FIXED INIT(0),  (X,DELMIN) BIN(53),
     FUN ENTRY(BIN(53),(*)BIN(53),(*)BIN(53)),
     (DELOUT,DELT,ER,FINTIM) BIN(53) EXT,  (I,J,N) FIXED;
 PUT EDIT('X',('Y',I DO I=1 TO N),'H')(SKIP,X(7),A(16),
     (N)(A,F(2),X(12)),A);
   PUT EDIT('ACCUM. STEPS', 'NO.OF STEPS','RATIO')
    (X(7),A(12),X(3),A(15),X(3),A(5));
      PUT SKIP(2);
  PUT EDIT(XO,(YO(I) DO I=1 TO N))(X(3),E(12,5));
 DELMIN=(FINTIM-XO)*1E-7;  DELSUM=DELOUT;  H=DELT;  EA=ER;
 X=XO;  Y=YO;  IT=-1;
      IIIT = -1;
 PR:  X=X+H;  IT=IT+1;
    IIIT = IIIT + 1;
     DO WHILE(X>DELSUM);    IR=(DELSUM-X+H)/H;  Y3=Y*IR+YOLD*(1-IR);
```

```
              PUT SKIP EDIT(DELSUM,(Y3(I) DO I=1 TO N),H)(X(3),E(12,5)); INTE 360
               PUT EDIT(IITT) (X(5), F(6));                              INTE 370
    IF IIT>0 THEN                                                        INTE 380
              PUT EDIT(IT,100*IT/IIT)                                    INTE 390
                                         (X(6),F(3),X(7),F(3));          INTE 400
              DELSUM=DELSUM+DELOUT;  IT,IIT=0;                           INTE 410
     END;                                                                INTE 420
     IF EA < 1.E-15 THEN EA = ER/5. ;                                    INTE 430
             H=H*0.9*(ER/EA)**0.25;                                      INTE 440
 INTEG:  IF X>FINTIM THEN GOTO OUT;              IIT=IIT+1;              INTE 450
           IITT = IITT + 1;                                              INTE 460
       CALL FUN(X,Y,K1);  K1=H*K1;                                       INTE 470
       CALL FUN(X+H/2,Y+K1/2,K2);  K2=H*K2;                              INTE 480
       CALL FUN(X+H/2,Y+K2/2,K3);  K3=H*K3;                              INTE 490
       CALL FUN(X+H,Y-K1+2*K2,K33);  K33=H*K33;                          INTE 500
       CALL FUN(X+H,Y+K3,K4);  K4=H*K4;                                  INTE 510
       YOLD=Y;  Y=YOLD+(K1+2*K2+2*K3+K4)/6;                              INTE 520
       Y3=YOLD+(K1+4*K2+K33)/6;                                          INTE 530
       EA=SQRT(SUM((Y(*)-Y3(*))**2));   IF EA<ER THEN GOTO PR;           INTE 540
      IF H>DELMIN THEN DO;                                               INTE 550
     IF EA < 1.E-15 THEN EA = ER/5. ;                                    INTE 560
                          H=H*0.9*(ER/EA)**.25;                          INTE 570
              GOTO INTEG;                                                INTE 580
       END;  ELSE DO; PUT EDIT('TOO SMALL STEP SIZE REQUIRED,INTEGRATION INTE 590
 TERMINATED')(SKIP(5),A);  GOTO OUT;  END;                               INTE 600
 OUT: END RUNGE;                                                         INTE 610
```

```
STMT LEVEL NEST
  1              FUN:   PROC (X,Y,K);
  2     1        DCL (X,(Y,K)(*))BIN(53);
  3     1        DCL DK BIN(53) INIT(3.E00);
  4     1        DCL EPSLON BIN(53) INIT(0.010204081632E0);
  5     1          K(1)  = - Y(1) - Y(1)*Y(2) + EPSLON*DK*Y(2)    ;
  6     1          K(2) = ( Y(1) - Y(1)*Y(2) - EPSLON*DK*Y(2))/EPSLON    ;
  7     1        END FUN;
```

PROBLEM 3.5 THE THERMAL DECOMPOSITION OF OZONE
0
1
0

DELOUT=1.
DELT=1E-4
ER=1E-5
FINTIM=100;

X	Y 1	Y 2	H	ACCUM. STEPS	NO.OF STEPS	RATIO
0.00000E+00	1.00000E+00	0.00000E+00				
1.00000E+00	1.59940E-01	8.50221E-01	2.75445E-02	160	80	100
2.00000E+00	3.27231E-02	5.96963E-01	4.87246E-02	216	28	100
3.00000E+00	1.62147E-02	3.81748E-01	8.19751E-02	248	16	100
4.00000E+00	9.55009E-03	2.60172E-01	1.29946E-01	268	10	100
5.00000E+00	6.66128E-03	1.92722E-01	1.79822E-01	280	6	100
6.00000E+00	5.08784E-03	1.51830E-01	2.38824E-01	290	5	100
7.00000E+00	4.10486E-03	1.24802E-01	2.99558E-01	298	4	100
8.00000E+00	3.43673E-03	1.05785E-01	3.53408E-01	304	3	100
9.00000E+00	2.95122E-03	9.16469E-02	4.13487E-01	310	3	100
1.00000E+01	2.58768E-03	8.08786E-02	4.55685E-01	314	2	100
1.10000E+01	2.30207E-03	7.23160E-02	4.97861E-01	318	2	100
1.20000E+01	2.07258E-03	6.53696E-02	5.37900E-01	322	2	100
1.30000E+01	1.88403E-03	5.96184E-02	5.73517E-01	326	2	100
1.40000E+01	1.72663E-03	5.47870E-02	5.89086E-01	328	1	100
1.50000E+01	1.59428E-03	5.07023E-02	6.15221E-01	332	2	100
1.60000E+01	1.47981E-03	4.71544E-02	6.34993E-01	336	2	100
1.70000E+01	1.38100E-03	4.40800E-02	6.42832E-01	338	1	100
1.80000E+01	1.29438E-03	4.13761E-02	6.55247E-01	342	2	100
1.90000E+01	1.21786E-03	3.89806E-02	6.60122E-01	344	1	100
2.00000E+01	1.14998E-03	3.68502E-02	6.67833E-01	348	2	100
2.10000E+01	1.08910E-03	3.49353E-02	6.70877E-01	350	1	100
2.20000E+01	1.03442E-03	3.32119E-02	6.75753E-01	354	2	100
2.30000E+01	9.84894E-04	3.16480E-02	6.77707E-01	356	1	100
2.40000E+01	9.39884E-04	3.02245E-02	6.80873E-01	360	2	100
2.50000E+01	8.98829E-04	2.89241E-02	6.82161E-01	362	1	100
2.60000E+01	8.61119E-04	2.77280E-02	6.84291E-01	366	2	100
2.70000E+01	8.26550E-04	2.66302E-02	6.85172E-01	368	1	100
2.80000E+01	7.94491E-04	2.56110E-02	6.86644E-01	372	2	100
2.90000E+01	7.64992E-04	2.46721E-02	6.87262E-01	374	1	100
3.00000E+01	7.37407E-04	2.37933E-02	6.87819E-01	376	1	100
3.10000E+01	7.11938E-04	2.29812E-02	6.88771E-01	380	2	100
3.20000E+01	6.88016E-04	2.22178E-02	6.89176E-01	382	1	100
3.30000E+01	6.65743E-04	2.15064E-02	6.89876E-01	386	2	100
3.40000E+01	6.44803E-04	2.08371E-02	6.90184E-01	388	1	100
3.50000E+01	6.25159E-04	2.02088E-02	6.90723E-01	392	2	100
3.60000E+01	6.06678E-04	1.96173E-02	6.90957E-01	394	1	100
3.70000E+01	5.89225E-04	1.90584E-02	6.91370E-01	398	2	100
3.80000E+01	5.72794E-04	1.85319E-02	6.91557E-01	400	1	100
3.90000E+01	5.57186E-04	1.80315E-02	6.91893E-01	404	2	100
4.00000E+01	5.42484E-04	1.75600E-02	6.92039E-01	406	1	100
4.10000E+01	5.28468E-04	1.71102E-02	6.92175E-01	408	1	100
4.20000E+01	5.15210E-04	1.66845E-02	6.92427E-01	412	2	100
4.30000E+01	5.02565E-04	1.62783E-02	6.92545E-01	414	1	100
4.40000E+01	4.90540E-04	1.58919E-02	6.92751E-01	418	2	100
4.50000E+01	4.79074E-04	1.55234E-02	6.92842E-01	420	1	100
4.60000E+01	4.68117E-04	1.51710E-02	6.93019E-01	424	2	100
4.70000E+01	4.57674E-04	1.48350E-02	6.93103E-01	426	1	100

4.80000E+01	4.47650E-04	1.45125E-02	6.93249E-01	430	2	100
4.90000E+01	4.38099E-04	1.42050E-02	6.93314E-01	432	1	100
5.00000E+01	4.28909E-04	1.39091E-02	6.93379E-01	434	1	100
5.10000E+01	4.20124E-04	1.36261E-02	6.93508E-01	438	2	100
5.20000E+01	4.11674E-04	1.33518E-02	6.93566E-01	440	1	100
5.30000E+01	4.03562E-04	1.30924E-02	6.93664E-01	444	2	100
5.40000E+01	3.95766E-04	1.28410E-02	6.93714E-01	446	1	100
5.50000E+01	3.88253E-04	1.25988E-02	6.93816E-01	450	2	100
5.60000E+01	3.81037E-04	1.23660E-02	6.93860E-01	452	1	100
5.70000E+01	3.74061E-04	1.21410E-02	6.93934E-01	456	2	100
5.80000E+01	3.67363E-04	1.19248E-02	6.93973E-01	458	1	100
5.90000E+01	3.60879E-04	1.17156E-02	6.94017E-01	460	1	100
6.00000E+01	3.54633E-04	1.15139E-02	6.94092E-01	464	2	100
6.10000E+01	3.48592E-04	1.13189E-02	6.94121E-01	466	1	100
6.20000E+01	3.42754E-04	1.11303E-02	6.94182E-01	470	2	100
6.30000E+01	3.37112E-04	1.09480E-02	6.94219E-01	472	1	100
6.40000E+01	3.31643E-04	1.07714E-02	6.94281E-01	476	2	100
6.50000E+01	3.26361E-04	1.06007E-02	6.94302E-01	478	1	100
6.60000E+01	3.21230E-04	1.04349E-02	6.94324E-01	480	1	100
6.70000E+01	3.16274E-04	1.02746E-02	6.94385E-01	484	2	100
6.80000E+01	3.11456E-04	1.01189E-02	6.94416E-01	486	1	100
6.90000E+01	3.06789E-04	9.96796E-03	6.94453E-01	490	2	100
7.00000E+01	3.02257E-04	9.82140E-03	6.94470E-01	492	1	100
7.10000E+01	2.97855E-04	9.67902E-03	6.94526E-01	496	2	100
7.20000E+01	2.93585E-04	9.54086E-03	6.94553E-01	498	1	100
7.30000E+01	2.89426E-04	9.40632E-03	6.94581E-01	502	2	100
7.40000E+01	2.85395E-04	9.27586E-03	6.94594E-01	504	1	100
7.50000E+01	2.81464E-04	9.14864E-03	6.94618E-01	506	1	100
7.60000E+01	2.77648E-04	9.02513E-03	6.94671E-01	510	2	100
7.70000E+01	2.73928E-04	8.90473E-03	6.94684E-01	512	1	100
7.80000E+01	2.70310E-04	8.78757E-03	6.94701E-01	516	2	100
7.90000E+01	2.66785E-04	8.67345E-03	6.94723E-01	518	1	100
8.00000E+01	2.63348E-04	8.56215E-03	6.94773E-01	522	2	100
8.10000E+01	2.60004E-04	8.45384E-03	6.94782E-01	524	1	100
8.20000E+01	2.56736E-04	8.34798E-03	6.94794E-01	528	2	100
8.30000E+01	2.53558E-04	8.24504E-03	6.94815E-01	530	1	100
8.40000E+01	2.50451E-04	8.14438E-03	6.94843E-01	532	1	100
8.50000E+01	2.47423E-04	8.04628E-03	6.94869E-01	536	2	100
8.60000E+01	2.44465E-04	7.95044E-03	6.94868E-01	538	1	100
8.70000E+01	2.41577E-04	7.85685E-03	6.94897E-01	542	2	100
8.80000E+01	2.38758E-04	7.76549E-03	6.94925E-01	544	1	100
8.90000E+01	2.36001E-04	7.67611E-03	6.94945E-01	548	2	100
9.00000E+01	2.33311E-04	7.58893E-03	6.94940E-01	550	1	100
9.10000E+01	2.30677E-04	7.50352E-03	6.94947E-01	552	1	100
9.20000E+01	2.28107E-04	7.42019E-03	6.94999E-01	556	2	100
9.30000E+01	2.25589E-04	7.33856E-03	6.95016E-01	558	1	100
9.40000E+01	2.23128E-04	7.25877E-03	6.95005E-01	562	2	100
9.50000E+01	2.20720E-04	7.18068E-03	6.95011E-01	564	1	100
9.60000E+01	2.18362E-04	7.10421E-03	6.95067E-01	568	2	100
9.70000E+01	2.16057E-04	7.02943E-03	6.95081E-01	570	1	100
9.80000E+01	2.13799E-04	6.95608E-03	6.95062E-01	574	2	100
9.90000E+01	2.11586E-04	6.88439E-03	6.95068E-01	576	1	100
1.00000E+02	2.09417E-04	6.81404E-03	6.95097E-01	578	1	100

IHE140I FILE SYSIN - END OF FILE ENCOUNTERED IN STATEMENT 00004 AT OFFSET +00094 FROM ENTRY POINT INTEG

APPENDIX E

This appendix contains:

1) Listing of FSTIME routine.

2) Usage of such a subroutine to determine the integration time for the "Mathematical model of human muscle-chemical aspects" problem with an error allowance of 1×10^{-3}.

```
*    FSTIME                                                           FSTI  10
FSTIME   CSECT                                                        FSTI  20
*          FAST TIME SUBROUTINE FOR IBM 360/44 WITH THE               FSTI  30
*          HIGH RESOLUTION TIMER FEATURE.                             FSTI  40
*      USSAGE:                                                        FSTI  50
*          CALL FSTIME(K1)                                            FSTI  60
*                                                                     FSTI  70
*                K1 = AN INTEGER*4 VARIABLE IN WHICH THE CLOCK VALUE IS FSTI  80
*                     RETURNED.                                       FSTI  90
*                                                                     FSTI 100
*          NOTE:                                                      FSTI 110
*                IN O. S. OPERATING SYSTEM THE INTERNAL CLOCK IS RESET AT FSTI 120
*                6:00 O'CLOCK AND AT MIDNIGHT.  FSTIME DOES NOT CHECK FSTI 130
*                FOR THIS,  SO DO NOT USE FSTIME AT THESE TIMES.      FSTI 140
*                                                                     FSTI 150
*          NOTE 2:                                                    FSTI 160
*                THE CLOCK IS A DOWN COUNTER THAT IS DECREMENTED      FSTI 170
*          EVERY 13.002 MICRO SECONDS BY A 76.8 KC OSC.  SINCE THE    FSTI 180
*          CLOCK IS A DOWN COUNTER, ONE MUST SUBTRACT THE SECOND TIME FSTI 190
*          FROM THE FIRST TIME TO GET A POSITIVE TIME DIFFERENCE.     FSTI 200
*          EXAMPLE:                                                   FSTI 210
*                CALL FSTIME(K1)                                      FSTI 220
*                CALL FSTIME(K2)                                      FSTI 230
*                KTOTAL = K1 - K2                                     FSTI 240
*                TIME = KTOTAL * .000013                              FSTI 250
*          THE TIME DIFFERENTIAL IS IN 'TIME' IN MICRO SECONDS.       FSTI 260
*                                                                     FSTI 270
*                           EXECUTION TIME -  MICRO SECONDS           FSTI 280
           L      1,0(1)    2.25                                      FSTI 290
           L      0,80      2.0                                       FSTI 300
           ST     0,0(1)    2.25                                      FSTI 310
           BCR    15,14     1.75                                      FSTI 320
*                         ---------                                   FSTI 330
*                TOTAL TIME =8.25                                     FSTI 340
           END                                                        FSTI 350
```

```
          C     -------------------------------------------------------------------
          C     -------------------------------------------------------------------
          C
0001              IMPLICIT REAL*8(A-H,O-Z)
0002              REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMO2,MTMCO2,KCSHM
0003              REAL*4 PRIRA
          C
0004              COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,FB,ABN2,ATN2
0005              COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
0006              COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
0007              COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PC2MO,PSCM,HCO3SM,HSM,
                 1PCCM,HCC3CM,HCM,PSOM,PSOMO,PCCM,PCCMC,PN2M,PN2MC,PSNM,PSNMO,PCNM,
                 2PCNMC
0008              COMMON/BLOCK5/YM(8,11)
0009              COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSOM,DSCOM,
                 1   QM,VMB,CMMAX,DPSNM,DSCNM,MTMC2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,
                 2   DSPHM,DCSBM,KCSHM,FM
0010              COMMON/BLOCK8/FFTIM
0011              COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL
0012              COMMON/BLCK10/J
0013              COMMON/BLCK11/T
0014              COMMON/BLCK12/DELT
0015              COMMON NOFNS
          C
0016              EXTERNAL DIFFUN,PEDERV,CUTP
          C
          C     -------------------------------------------------------------------
          C
          C
0017              WRITE(6,610)
0018          610 FORMAT(1H1,/1X,36('*')/
                 |             1X,'*PROBLEM 3.7     A BIOLOGICAL SYSTEM*'/
                 | 1X,36('*')//)
0019              CALL DATA
          C
0020              TIME=0.0D00
0021              SEL = 16.0D00
0022              FTIME=16.0D00
0023              FFTIM=2.05D00
0024              DELT=0.05D00
0025              PRIRA = SEL/DELT
0026              KCS = 0
0027              T=0.0D00
0028              SPLM=0.0D00
          C
          C
          C     -------------------------------------------------------------------
          C
          C*******************************************************************
          C*        TAKE THE READING OF THE TIMER BEFORE STARTING INTEGRATION    *
          C-------------------------------------------------------------------
          C
0029              CALL FSTIME(KRT1)
          C
```

```
0030      5     IF(SPLM .EQ. 0.0D00) CALL INTAL                                  ETSS 450
0031            SPLM=SPLM+DELT                                                   ETSS 460
0032            IF(SPLM .GT. FTIME) GO TO 40                                     ETSS 470
0033            IF(DABS(SPLM-FFTIM) .LE. 0.0001D00) GO TO 2                      ETSS 480
0034            GO TO 1                                                          ETSS 490
          C                                                                      ETSS 500
0035      2     CONTINUE                                                         ETSS 510
0036            CALL FORCE                                                       ETSS 520
          C                                                                      ETSS 530
0037      1     CONTINUE                                                         ETSS 540
0038            IF(KOS .LT. IFIX(4.E00*PRIRA))GO TO 11
0039            KOS = 0
0040            WRITE(6,215)
0041       215  FORMAT(1H1///)
0042        11  CONTINUE
0043            KOS = KOS + 1
          C                                                                      ETSS 550
          C     -----------------------------------------------------------------ETSS 560
          C                                                                      ETSS 570
0044            CALL MUSCLE                                                      ETSS 580
          C                                                                      ETSS 590
          C     -----------------------------------------------------------------ETSS 600
          C                                                                      ETSS 610
0045            GO TO 5                                                          ETSS 620
          C                                                                      ETSS 630
0046        40  CONTINUE                                                         ETSS 640
          C*****************************************************************
          C*        TAKE THE READING OF THE TIMER AFTER THE INTEGRATION IS OVER *
          C-----------------------------------------------------------------
0047            CALL FSTIME(KRT2)
          C
          C*****************************************************************
          C*        CALCULATE THE ELAPSED TIME AND PRINT IT OUT                 *
          C-----------------------------------------------------------------
          C
0048            KTOTAL = KRT1 - KRT2
0049            TIMEI= KTOTAL*13.02D-06
0050            WRITE(6,5555)TIMEI
0051      5555 FORMAT(///20X,'THE INTEGRATION TIME IS : ', D15.7, 2X, 'SECONDS')
          C
0052            STOP                                                             ETSS 700
0053            END                                                              ETSS 710
```

```
0001          SUBROUTINE MUSCLE
        C
        C    -----------------------------------------------------------------------
        C
        C
0002          IMPLICIT REAL*8(A-H,O-Z)
0003          REAL*8 KPRIME,KA,KC,KU,KV,K,KZ,KO,MTMC2,MTMCO2,KCSHM,MTMMO2
0004          REAL*8 INTAMA,INTAMM,MM,MMS,MMC
0005          REAL*8 L1,L2
0006          REAL*4 PW,PPW
        C
        C    -----------------------------------------------------------------------
        C
0007          COMMON/BLOCK1/HCRIT,BETAP,BETAE,R,CMAX,ALPHAC,ALPHAO,HB,ABN2,ATN2
0008          COMMON/BLOCK2/KPRIME,KA,KC,KU,KV,K,KZ,KO
0009          COMMON/BLOCK3/PCO2AM,HCO3AM,CARBAM,PO2AM,CO2AM,PN2AM
0010          COMMON/BLOCK4/PCO2M,HCO3PM,CARBM,PO2M,CO2M,PO2MO,PSCM,HCO3SM,HSM,
             1PCCM,HCO3CM,HCM,PSOM,PSOMC,PCCM,PCCMC,PN2M,PN2MC,PSNM,PSNMC,PCNM,
             2PCNMC
0011          COMMON/BLOCK5/YM(8,11)
0012          COMMON/BLOCK7/BETASM,BETACM,VSM,VCM,DSPCM,DCSCM,DPSCM,DSCOM,
             1   CM,VMB,CMMAX,DPSNM,DSCNM,MTMO2,MTMCO2,TCSM,RCSM,RSPM,DSPBM,
             2   DSPHM,DCSBM,KCSHM,FM
0013          COMMON/BLOCK8/FFTIM
0014          COMMON /BLOCK9/ FSEL,SPLM,DSEL,SEL
0015          COMMON/BLOCK10/J
0016          COMMON/BLCK11/T
0017          COMMON/BLOK12/DELT
0018          COMMON NOFNS
        C
0019          DIMENSION SAVE(12,11),YMAX(11),YF(11),ERROR(11),PW(121),
             1          PPW(121),LLL(11),MMM(11)
        C
0020          EXTERNAL DIFFUN,PEDERV,OUTP
        C
        C    -----------------------------------------------------------------------
        C
        C
0021          IF(T    .EQ. 0.0000) GO TO 111
0022          GO TO 222
0023    111   NOFNS=0
0024          J=0
0025          JSTART=0
        C
        C       PARAMETERS FOR GEAR ' S  SYSTEM
        C
0026          FSEL=SEL
0027          N=11
        C
        C     PARTIAL DERIVATIVES PROVIDED ANALYTICALLY IN PEDERV
        C
0028          MF = 1
        C
0029          HMIN=1.0D-10
```

```
0030            EPS = 1.0D-03
0031            MAXDER=6
0032            HMAX=DELT
0033            DSEL=DELT
0034            H=1.0D-04
0035            DELTA = DSEL
0036            DO 10 I=1,11
0037            YMAX(I)=1.0D00
0038      10    CONTINUE
0039      222   CONTINUE
          C
0040            IF(T .GT. 0.0D00) H=DELT
0041            IF(DABS(T-FFTIM) .LE. DELT) H=1.0D-04
          C
          C    -------------------------------------------------------------
          C
0042            CALL MPGRM2(N,T,YM,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,ERROR,KFLAG,
               1JSTART,MAXDER,PW,PPW,LLL,MMM,YF,DELTA,DIFFUN,PEDERV,OUTP)
          C
          C    -------------------------------------------------------------
          C
          C    -------------------------------------------------------------
          C
0043      100   CONTINUE
          C
0044            RETURN
0045            END
```

```
**********************************
*PROBLEM 3.7    A BIOLOGICAL SYSTEM*
**********************************

------------------------------------------------------------------------------------------------------------------------------

NUMERICAL INTEGRATION SOL.  (GEAR)  :

TIME =     16.000      STEP SIZE = 0.50000D-01     DE. EV. =  1014     ACCUM. STEPS =   969      H  = 0.10957D-01
                       PCO2M =  0.486527500457D 02     HCO3PM =  0.262296099929D-01
                       PSCM  =  0.501107993446D 02     HCO3SM =  0.282125520641D-01     HSM =  0.421144301555D-07
                       PCCM  =  0.539740313150D 02     HCO3CM =  0.121689597648D-01     HCM =  0.105661376585D-06
                       PO2M =  0.322917277257D 02     PSOM =  0.273206325177D 02     PCOM =  0.124066402480D 02

                     THE INTEGRATION TIME IS :   0.636993 7D 02  SECONDS
```