Eldon Hansen

6-90-64-6 • FEBRUARY 1964

6-90-64-6

TECHNICAL REPORT: MATHEMATICS

DIFEQ INTEGRATION ROUTINE - USER'S MANUAL

Lockheed

MISSILES & SPACE COMPANY

A GROUP DIVISION OF LOCKHEED AIRCRAFT CORPORATION
SUNNYVALE, CALIFORNIA

LMSC 6-90-64-6

February 18, 1964

Technical Report: Mathematics

DIFEQ INTEGRATION ROUTINE - USER'S MANUAL

by

R. E. Moore

J. A. Davison

H. R. Jaschke

S. Shayer

Lockheed Missiles & Space Company
Research Laboratories
Palo Alto, California

CONTENTS

Section 1
INTRODUCTION

It is the purpose of this paper to describe the use of a novel program, DIFEQ, written for the IBM 7094 computer for "solving" systems of ordinary differential equations, that is to say, for finding numerical values of particular solutions with assigned initial conditions.

The instructions for the use of the program are supposed to be self-contained and enable the reader to use the program with little or no programming experience.

The program has several features which distinguish it from other integration routine:

1. In addition to providing approximate solution values, the program supplies a rigorous upper bound on the total error of each solution component at each computed point. The user may think of the results as having the form $Y \pm e$ where Y is the approximate solution and e is the error bound. If y is the exact solution at the given point, then $| y - Y | \leq e$ holds.

2. In order to use the routine, all the user has to supply is his differential equations and initial conditions. The program itself determines all the intrinsic parameters such as initial and subsequent step sizes, etc. If the user desires, he may specify values of the independent variable at which he wishes solution values; otherwise the program will even select these. In case he leaves the choice of output points up to the program, the first nine Taylor coefficients

will be printed, along with solutions values, for interpolating intermediate points. Instructions for the use of these coefficients will appear with the output.

The routine is applicable immediately to any system of ordinary differential equations which can be written

$$\frac{dy_i}{dx} = F_i(y_1, y_2, \ldots, y_m) \quad i = 1, 2, \ldots, m$$

where the functions F_i are rational in their arguments. There is no loss of generality in assuming that the independent variable x is missing from the arguments since the substitution

$$y_{m+1} = x$$

and the addition of the equation

$$\frac{dy_{m+1}}{dx} = 1$$

can be made in order to remove the variable x from explicit occurrence in the functions F_i should it so occur.

Square roots are also permitted to appear explicitly in the functions F_i.

In order to apply the routine to the solution of differential systems in which functions other than rational ones and square root occur explicitly, the system must first be rewritten in such a way that those occurrences are made implicit.

A large number of functions commonly occurring in the physical sciences satisfy rational differential systems themselves and can thus be replaced for our purposes by their defining differential equations. For example, sin and cos satisfy

$$\ddot{y} + y = 0 \qquad \text{or } \dot{y}_1 = y_2 \text{ , } \dot{y}_2 = -y_1$$

and the function $\tan^{-1}$ satisfies

$$y' = 1/(1 + x^2)$$

and exp satisfies

$$y' = y$$

etc., etc., etc.

As an example, we can reduce the pendulum equation

$$\ddot{y} + \sin y = 0$$

to a rational system as follows. First reduce the second order equation to a pair of first order equations.

Substitute

$$y_1 = y$$

$$y_2 = \dot{y}$$

then

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = \ddot{y} = -\sin y_1$$

We must still remove the non-rational function $\sin y_1$ from explicit occurrence. We define a new variable

$$y_3 = \sin y_1 \quad ;$$

then

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -y_3$$

But now we must add another equation defining $\dot{y}_3$. We have

$$\dot{y}_3 = (\cos y_1)\, \dot{y}_1 = (\cos y_1)\, y_2$$

Again we substitute

$$y_4 = \cos y_1$$

and write

$$\dot{y}_4 = -(\sin y_1)\, \dot{y}_1 = -y_3\, y_2$$

Thus we have replaced the original 2nd order equation

$$\ddot{y} + \sin y = 0$$

by the following system of four first order equations involving only rational functions

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = -y_3$$

$$\dot{y}_3 = y_4 \; y_2$$

$$\dot{y}_4 = -y_3 \; y_2$$

We recall that

$$y_1 = y$$

$$y_2 = \dot{y}$$

$$y_3 = \sin \; y_1$$

$$y_4 = \cos \; y_1$$

Thus if our original initial conditions were $y(t_0)$, $\dot{y}(t_0)$, we now use

$$y_1(t_0) = y \; (t_0)$$

$$y_2(t_0) = \dot{y} \; (t_0)$$

$$y_3(t_0) = \sin \; y(t_0)$$

$$y_4(t_0) = \cos \; y(t_0)$$

3. In specifying initial conditions and equation constants, inexact data is allowed; i.e., data of the form, $x \pm e$. These initial errors will also be taken into account by the program.

Section 2

HOW TO USE DIFEQ

A. Reduction of the differential equations to "basic steps"

1. Reduce the sytem of differential equations to be solved to a system of "m" first order equations, $m \leq 20$. The system so obtained must give an explicit expression for the first derivative of each variable. These expressions may be any sequence of constants and variables connected by the following operations: addition, subtraction, multiplication, inverse, and square root. Each division must be rewritten as a reciprocal, or "inverse", and a product; for example,

$$U/V \text{ is to be rewritten}$$

$$U\,(V)^{-1} \quad \text{or} \quad (V)^{-1}\,U \quad .$$

Example 1):

The equation

$$y'' + 3yy' + b\,(y^2 - 1)^{1/2} = 0$$

must be rewritten in a form such as the following:

Let $y = y_1$, $y' = y_2$,

then

$$y_1' = y_2$$

$$y_2' = -\,3y_1\,y_2 - b\,(y_1^2 - 1)^{1/2}$$

Example 2):

If we wish to apply the routine to the equation

$$y' = y^2 + (xy)^{1/2}$$

in which the independent variable "x" occurs explicitly, we must add the equation

$$x' = 1 .$$

Example 3):

The equations

$$y_1' = 2 - y_2$$
$$y_2' = y_2/ (b + y_1)^{1/2}$$

are almost in proper form as they are originally stated. All we need to do is rewrite the division in the second one as

$$y_2' = y_2 ((b + y_1)^{1/2})^{-1}$$

2. After having put the system of equations to be solved into the form just described, the next step is to revise the notation to correspond with that acceptable to the routine DIFEQ.

Rename all variables occurring in the expressions for the derivatives so that each variable is denoted by Y(i,0), for some i = 1, 2, ... , m. The first derivatives of the Y(i,0) are denoted by Y(i,1). A maximum value of i = m in a particular set of equations requires that a unique definition be given for each Y(i,1), i = 1, 2, ... , m. Constants are denoted by CK CΦNSTANT where the maximum value for K is 39; for example, we might have the constants C1 CΦNSTANT, C2 CΦNSTANT. If the user's constants include any of the integers 0, 1, 2, ... , 10, he may specify them by their alphabetic names together with the word "CΦNSTANT"; i.e., ZERΦ CΦNSTANT, ΦNE CΦNSTANT, ... , TEN CΦNSTANT.

Example:

For the last example given above, the revised notation is:

Y(1, 1) = TWØ CØNSTANT - Y(2,0)
Y(2,1) = Y(2,0) ((C1 CØNSTANT + Y(1,0) $)^{1/2})^{-1}$

3. In order to fulfill the requirements of DIFEQ a further reduction of the system is necessary. This final reduction yields basic steps of one operation each, with one or two arguments (depending on the operation), and one result per step. The user may thus have to define one or more intermediate results for a given derivative.

These intermediate results are to be denoted by T(j,0) for j = 1, 2, ... , n; n ≤ 200. Each T(j,0) must be uniquely defined by an explicit expression, and a maximum value j = n requires that a unique definition be given each T(j,0); j = 1, ... , n.

There are six operations allowed in basic steps:

addition:	ADD p + q = T(j,0)
subtraction:	SUB p - q = T(j,0)
multiplication:	MULT p*q = T(j,0)
inverse:	FIND INVERSE OF p = T(j,0)
square root:	SQROOT p IS T(j,0)
defining a derivative:	DEFINE p AS Y(i,1)

The arguments p,q in the basic steps may be the names of constants, e.g., C3 CØNSTANT, variables denoted by Y(i,0), or intermediate results denoted by T(j,0).

Note the following restrictions on the use of the T(j,0):

1) A given T(j,0) must appear as the result of a basic step,(i.e., on the right hand side of an equation), prior to its appearance as an argument, (i.e., on the left hand side of an equation).

2) No T(j,0) may appear on both sides of a given equation.

3) No T(j,0) may appear on the right hand side of more than one equation.

Each step is to be punched into an IBM card, with one step per card. The following example shows the spacing which should be used in punching the equation cards. This is then the final form of the last example mentioned above.

The first letter may be punched in any column from 8 through 15 inclusive. One blank column must appear between the words and symbols except no blanks should be between the "=" sign and the symbols on either side.

```
SUB TWØ CØNSTANT - Y(2,0)=T(1,0)
DEFINE T(1,0) AS Y(1,1)
ADD C1 CØNSTANT + Y(1,0)=T(2,0)
SQRØØT T(2,0) IS T(3,0)
 FIND INVERSE ØF T(3,0)=T(4,0)
MULT Y(2,0) * T(4,0)=T(5,0)
DEFINE T(5,0) AS Y(2,1)
```

4. Preceding the equation cards an additional card is required. It has the format:

INITIALIZE M = ______, N = ______, C =_______

where M is the number of variables of the form Y(i,0), N is the number of intermediate results T(j,0), and C is the number of constants CK. The value of C does <u>not</u> include any of the integer constants written by names other than CK CØNSTANT, e.g., ØNE CØNSTANT. The first letter of INITIALIZE may be punched in any column from 8 through 15. One blank column must appear between the word INITIALIZE and the letter M. No blanks should be used in the remainder of the statement. The number of equation cards will always be exactly equal to M+N, if correctly prepared.

For the equation deck in the above example we would have:

INITIALIZE M=2,N=5,C=1

Additional examples illustrating the preparation of the differential equation cards are given toward the end of this manual.

Next we will describe the preparation of initial conditions on punched cards.

B. Input Data

The input data deck consists of data for one or more sets of initial conditions and parameter values for a given set of equations. Each set is composed of five types of data. We may think of each of the types of data as comprising a card group.

Card Group I. consists of any number of cards. An "*" must be the first symbol punched on each of these cards. On the remainder of each card may be punched any information which is desired as an output title. This might include such items as user's name, problem title, date, and case number.

Card Group II. gives the initial conditions for the variables. An initial value must be assigned to each Y(i,0) used in the "equation deck". Each initial value Y(i,0) is identified by punching "Yi=" to the left of the value, e.g., "Y2=" would identify the initial value of Y(2,0). The value itself is given by a pair of numbers "y,ε", interpreted by the program as $y \pm \varepsilon$. The numbers "y" and "ε" are separated by a comma on the punched card. If exact data is used, i.e., "ε"=0; "ε" may be omitted if and only if the comma is also omitted. The numbers "y" and "ε" each have the form of a decimal number with a maximum of eight significant digits, and may be followed by an exponent if desired. If the decimal point is omitted, it is assumed to be at the right of the number. Values greater in absolute value than 10^{38} and values between zero and 10^{-38} cannot be handled by the program. An exponent is interpreted as multiplying the number by the given power of ten. Use of an exponent is indicated by punching an "E" after the number, then the sign of the exponent, finally its value.

Note: Only one"Yi= ..." may be punched per card.

Card Group III. gives the values of all constants used in the "equation deck". Each constant is identified by punching "Ck=",k=1,2,..C, to the left of the value. Values have the same form as those in Card Group II. Similarly, a separate card must be used for each Ck.

Card Group IV. (optional) consists of three cards. On the first card is punched the word "XZERϕ" or "XO" and a decimal number*,(fn.p.11), giving the initial value of the independent variable. The second card contains the word "XFINAL" or "XF" and the maximum value* to be attained by the independent variable*. The last card is identified by punching the word "PϕINTS" or PTS" onto the card and an integer indicating the maximum number of points at which the solution is to be evaluated. This number corresponds to the maximum number of integration steps computed by the program. Thus, it may have no relation to the number of solution points actually printed if the user selects a print option to control output. (see Card Group V.) Equal signs are optional in Card Group IV.

Card Group IV. or any part of it is optional. If omitted, the value of the independent variable associated with the initial conditions, XO, is assumed to be zero. The upper limit on the independent variable, XF, if omitted will be assumed to be 10^{38}, except in option (1) under Card Group V. where it is taken as the largest value of Xi given. If not specified, the maximum number of points allowed will be set to 32,767.

Card Group V. (optional) may be used to control program output.

(1) If the user desires solutions at specific predetermined values of the independent variable: X1 > X0 , X2 > X1, ... , XF; he should prepare a set of cards punched as follows:

X1= ...
X2= ...
X3= ...   etc.

The value of each Xi is to be punched as a single decimal number on a separate card. A maximum of 199 values may be used. These cards then constitute Card Group V.

(2) Should the user desire printed solutions at even increments of the independent variable, i.e., at

$$X0 + \Delta x;\ X0 + 2\Delta x,\ X0 + 3\Delta x,\ \ldots$$

he should punch a card with the word "DELTAX" and the decimal print increment value*, Δx. In this case Card Group V. will consist of one card only. Equal sign is optional.

(3) If Card Group V. is omitted, solutions will be printed at each integration step selected by the program.

* Values for "X0", "XF", "X1", "X2", ... , and "DELTAX" are to be punched as single decimal numbers according to the rules for punching a number "y" in Card Group II.

There are no restrictions as to specific column designations for the input data. However, the items on each card must appear in the prescribed order, and no extraneous information may appear on a data card. The ordering of the cards and card groups within a given set of data is also arbitrary, except that a card with the word "END" punched in it must follow each set of data.

(See examples)

C. Timing

The program requires a variable amount of time per integration step depending on such factors as number and complexity of equations and proximity to singularities at a given point. Another factor of uncertainty in estimating computation time is the number of integration steps required to obtain a solution at the maximum value of the independent variable. The timing experience on the sample problems given at the end of this manual may be helpful to the user.

If the solution is stopped by exceeding estimated run time, the user may input the last solution values printed and re-submit the program to continue integration. At all normal program exits a message will indicate computation time in minutes.

Section 3

METHOD

Solutions are obtained in a step by step fashion by means of expansions in Taylor series truncated at the ninth term. The remainder term in the Taylor series is bounded by the program over regions it constructs about each new solution point. The step size is chosen so that the solution remains in the region for all intermediate values between one solution point and the next. This containment is tested by the program.

Thanks to "interval arithmetic", (see Ref. 1), the program is able to bound the errors due to rounding of the finite precision machine arithmetic. (Single precision, about 8 decimal places, floating point arithmetic is used.) In fact, interval arithmetic is used throughout the computations and this is what enables the program to produce rigorously correct upper bounds on the overall error, even including error due to inexact initial conditions, conversion of decimal input to binary in the machine, etc.

Thanks to a compiler type program which is part of the DIFEQ routine and which makes use of a "macro-expander" program called XPOP, (see Ref.2), the required coding for the computation of values of the Taylor's coefficients is generated by the program itself. We have been able to do this in such a way that the computing time to get the nth Taylor coefficient goes up only linearly with n for all equations which can be reduced to the basic steps as described in the previous section of this manual. Due to this fact we are able to compute efficiently with nine terms of the Taylor's series.

A complete description of the program would be too long for inclusion in this manual. Reference 1 may be consulted for the main features of the methods used.

Section 4

EXAMPLES

Each of the following sample problems is traced through the preparation of the equation deck and input data deck.

Example 1:

Problem: $\frac{dy}{dx} = y^2;\ y(0) = 1.$

Step 1: $\frac{dy}{dx} = y \cdot y$

Step 2: Y(1,1) = Y(1,0) . Y(1,0)

Step 3: MULT Y(1,0) * Y(1,0)=T(1.0)

DEFINE T(1,0) As Y(1,1)

Thus, the equation deck should be punched as follows:

```
INITIALIZE M=1,N=1,C=0
MULT Y(1,0) * Y(1,0)=T(1,0)
DEFINE T(1,0) AS Y(1,1)
```

The input data deck may be punched:

```
*  EXAMPLE 1  DY/DX =Y**2
Y1 = 1.0
END
```

Example 2:

Problem: $\frac{d^2y}{dx^2} + y = 0;\ x_o = 0;$ ~~$x_f = 6.28$~~, $y(0) = 0,\ \frac{dy}{dx}(0) = 1$

Step 1: Let $y_1 = y$

$y_2 = \frac{dy}{dx}$

then $\dot{y}_1 = y_2$

$\dot{y}_2 = -y_1$

Step 2: Y(1,1) = Y(2,0)

Y(2,1) = -Y(1,0)

Step 3: Equation deck may be punched as a sequence of three basic steps:

```
INITIALIZE M=2,N=1,C=0
DEFINE Y(2,0) AS Y(1,1)
SUB ZERØ CØNSTANT - Y(1,0)=T(1,0)
DEFINE T(1,0) AS Y(2,1)
```

Input data deck:

```
*    SIN/CØS EQUATIØNS 1/15/64
Y1=0
Y2=1.
XZERO=0.
DELTAX=.5
XFINAL=9.00
END
```

An alternate input data deck:

```
* SIN/CØS EQUATIONS CASE 2 1/15/64
Y1=0
Y2=1.
XZERØ=0
X1=.123
X2=.234
X3=.345
X4=.456
X5=.567
X6=1.0
XF=1.57
```

Example 3:

Problem:
$$\ddot{x}_1 - 2\dot{x}_2 = x_1 - \frac{\mu(x_1 - 1 + \mu)}{r^3} - \frac{(1-\mu)(x_1 + \mu)}{R^3}$$

$$\ddot{x}_2 + 2\dot{x}_1 = x_2 - \frac{\mu x_2}{r^3} - \frac{(1-\mu)\, x_2}{R^3}$$

$$r = \{(x_1 - 1 + \mu)^2 + x_2^2\}^{1/2}$$

$$R = \{(x_1 + \mu)^2 + x_2^2\}^{1/2}$$

Step 1: Let $y_1 = x_1$

$$y_2 = x_2$$

$$y_3 = \dot{x}_1$$

$$y_4 = \dot{x}_2$$

then $r = \{(y_1 - 1 + \mu)^2 + y_2^2\}^{1/2}$

$$R = \{(y_1 + \mu)^2 + y_2^2\}^{1/2}$$

$$\dot{y}_1 = y_3$$

$$\dot{y}_2 = y_4$$

$$\dot{y}_3 = 2y_4 + y_1 - \frac{\mu(y_1 - 1 + \mu)}{r^3} - \frac{(1-\mu)(y_1 + \mu)}{R^3}$$

$$\dot{y}_4 = -2y_3 + y_2 - \frac{\mu y_2}{r^3} - \frac{(1-\mu)\, y_2}{R^3}$$

Step 2: Let μ be represented by C1 CØNSTANT. Reduce equations to steps of single operations, and punch cards with the following format:

```
INITIALIZE M=4,N=33,C=1
DEFINE Y(3,0) AS Y(1,1)
DEFINE Y(4,0) AS Y(2,1)
SUB Y(1,0) - ϕNE CϕNSTANT=T(1,0)
ADD T(1,0) + C1 CϕNSTANT=T(2,0)
MULT T(2,0) * T(2,0)=T(3,0)
MULT Y(2,0) * Y(2,0)=T(4,0)
ADD T(3,0) + T(4,0)=T(5,0)
SQRϕϕT T(5,0) IS T(6,0)                    [r]
ADD Y(1,0) + C1 CϕNSTANT=T(7,0)
MULT T(7,0) * T(7,0)=T(8,0)
ADD T(4,0) + T(8,0)=T(9,0)
SQRϕϕT T(9,0) IS T(10,0)                   [R]
SUB ϕNE CϕNSTANT - C1 CϕNSTANT=T(11,0)
MULT T(6,0) * T(6,0)=T(12,0)
MULT T(12,0) * T(6,0)=T(13,0)              [r^3]
FIND INVERSE ϕF T(13,0)=T(14,0)
MULT T(10,0) * T(10,0)=T(15,0)
MULT T(15,0) * T(10,0)=T(16,0)             [R^3]
FIND INVERSE ϕF T(16,0)=T(17,0)
MULT T(11,0) * T(7,0)=T(18,0)
MULT T(18,0) * T(17,0)=T(19,0)             [(1-μ)(x_1+μ) / R^3]
MULT C1 CϕNSTANT * T(2,0)=T(20,0)
MULT T(20,0) * T(14,0)=T(21,0)             [μ(x_1-1+μ) / R^3]
```

```
SUB Y(1,0) - T(21,0)=T(22,0)
SUB T(22,0) - T(19,0)=T(23,0)
MULT TWØ CØNSTANT * Y(4,0)=T(24,0)
ADD T(24,0) + T(23,0)=T(25,0)                [ẍ1]
DEFINE T(25,0) AS Y(3,1)
MULT C1 CØNSTANT * Y(2,0)=T(26,0)
MULT T(26,0) * T(14,0)=T(27,0)               [μx2/r^3]
MULT T(11,0) * Y(2,0)=T(28,0)
MULT T(28,0) * T(17,0)=T(29,0)               [(1-μ) x2/R^3]
MULT TWØ CØNSTANT * Y(3,0)=T(30,0)
SUB Y(2,0) - T(27,0)=T(31,0)
SUB T(31,0) - T(29,0)=T(32,0)
SUB T(32,0) - T(30,0)=T(33,0)                [ẍ2]
DEFINE T(33,0) AS Y(4,1)

*    1/27/64 3-BØDY PRØBLEM
  Y1=-1.98012E-02
  Y2=-1.50162E-02
  Y3=9.5560068
  Y4=-4.856878
  XZERØ=0.0

  C1=.01215
    END
```

Example 4: Illustrating reduction of non-rational systems

Problem: $$\frac{dy_1}{dx} = y_2 \cos(xy_1)$$

$$\frac{dy_2}{dx} = \exp(x^2 + 2x\ln y_1) = y_1^{2x} e^{x^2}$$

Let

$$y_3 = x$$

then $$\frac{dy_3}{dx} = 1$$

We must add variables and rational expressions for their first derivatives in order to obtain a first-order system in which the derivatives of all variables are defined in terms of the basic arithmetical operations allowed by the program. Thus we introduce the new variables

$$y_4 = y_3 y_1$$
$$y_5 = \ln y_1$$
$$y_6 = y_3^2 + 2y_3 y_5$$
$$y_7 = \cos y_4$$
$$y_8 = \exp(y_6)$$
$$y_9 = \sin y_4 .$$

Then using the facts that $(\ln y_1)' = (\frac{1}{y_1}) y_1'$; $(\cos y_4)' = -(\sin y_4) y_4'$; $(\sin y_4)' = (\cos y_4) y_4'$; $(\exp y_6)' = (\exp y_6) y_6'$; we can "reduce" the irrational system given for y_1, y_2 to the following rational system in the variables $y_1, y_2, y_3, \ldots, y_9$.

$y_1' = y_2 y_7$

$y_2' = y_8$

$y_3' = 1$

$y_4' = y_3 y_1' + y_1$

$y_5' = \frac{y_1'}{y_1}$

$y_6' = 2y_3 + 2y_3 y_5' + 2y_5$

$y_7' = -y_9 \; y_4'$

$y_8' = y_8 y_6'$

$y_9' = y_7 y_4'$

After the above equations are reduced to basic steps, the equation deck should be punched as follows:

```
INITIALIZE M=9,N=13,C=0
MULT Y(2,0) * Y(7,0)=T(1,0)
DEFINE T(1,0) AS Y(1,1)
DEFINE ϕNE CϕNSTANT AS Y(3,1)
MULT Y(3,0) * T(1,0)=T(2,0)
ADD T(2,0) + Y(1,0)=T(3,0)
DEFINE T(3,0) AS Y(4,1)
FIND INVERSE ϕF Y(1,0)=T(4,0)
MULT T(4,0) * T(1,0)=T(5,0)
DEFINE T(5,0) AS Y(5,1)
MULT Y(3,0) * T(5,0)=T(6,0)
ADD T(6,0) + Y(3,0)=T(7,0)
ADD T(7,0) + Y(5,0)=T(8,0)
```

```
MULT TWØ CØNSTANT * T(8,0)=T(9,0)
DEFINE T(9,0) AS Y(6,1)
MULT Y(9,0) * T(3,0)=T(10,0)
SUB ZERØ CØNSTANT - T(10,0)=T(11,0)
DEFINE T(11,0) AS Y(7,1)
MULT Y(8,0) * T(9,0)=T(12,0)
DEFINE T(12,0) AS Y(8,1)
MULT Y(7,0) * T(3,0)=T(13,0)
DEFINE T(13,0) AS Y(9,1)
DEFINE Y(8,0) AS Y(2,1)
```

Section 5

JOB DECK SET-UP

After preparing the equation deck and the input data deck the user will combine these two sets of cards with "DIFEQ program cards" and prepare the deck for submission to computer operations. It is assumed that the program will be run under the IBSYS monitor system. The "DIFEQ program cards" together with instructions for setting up the deck may be obtained from

R. E. Moore, DIFEQ
Dept. 52-20
Bldg. 201
Lockheed Missiles & Space Company
3251 Hanover Street
Palo Alto
California

Phone: 324-3311
Ext. 45417

Section 6

SAMPLE OUTPUT

The final pages of this manual contain sample output obtained from Examples 1, 2, and 3. Note that the form of print-out corresponds with the print option selected (Card Group V.) in each case.

In addition, the user should be aware that the following forms of output may occur:

(1) Illegal data will cause an error message to be printed, indicating the source of the error.

(2) The message "Error in above assembly" indicates the format of the punched equation cards is not compatible with program requirements.

(3) In case a numerical difficulty is encountered in obtaining a solution (such as overflow when the solution is approaching a singularity), the program will automatically attempt to overcome the difficulty by a "rescaling process". Should a certain number of trials fail, an error message will be printed indicating the last region over which the program was attempting to bound the remainder term in the Taylor's series. This error message is an indication of a singularity near the solution.

(4) In case the solution is a polynomial of degree less than or equal to eight, the program completes the solution in one step. The Taylor's expansion coefficients printed are valid for all values of the independent variable. In this case the user should ignore the error message.

We repeat once more that the actual error in the computed solution values is guaranteed to be less than the computed error bounds. In some cases the actual error will be considerably less than the computed error bound. In the sample output to follow, for the sin/cos equations the computed solution is found by comparison with published tables to have a maximum actual error of 6×10^{-8} at $x = 8.5$.

In example 1, the exact solution is $Y(1,0) = 1/(1-X)$ since $Y(1,1) = -(1/(1-X)^2)(-1) = Y(1,0)^2$.

Notice that the program stopped short of the singularity at X=1. Since the program rigorously bounds the exact solution even at points intermediate to those at which Taylor expansions are made, it can never go past a point where the solution becomes infinite. In most other integration routines such singularities can escape detection.

We have included only part of the output for example 1, namely the 1st, 2nd, 25th, 46th, and 87th computed points.

```
    EXAMPLE 1    DY/DX  =  Y**2
INPUT DATA
XZERO=  0.                  XFINAL=  1.7014100E 38      MAXIMUM NO. POINTS=32767
Y( 1,0)=  1.0000000E 00,        ERROR BOUND=  7.4505806E-09
FOR INTERMEDIATE VALUES, IF X IS BETWEEN X(Q) AND X(Q+1), SET X = X(Q) + T*DELTA
                  (WHERE T*DELTA IS T MULTIPLIED BY DELTA)
   DETERMINE T AND COMPUTE Y(P,0) AT (X(Q)+T*DELTA) = SUM OVER J OF (T TO THE POWER J TIMES Y(P,J))
   THIS VALUE WILL BE AT LEAST AS ACCURATE AS Y(P,0) AT X(Q+1).
```

```
SOLUTION AT X(   1)=  4.4199569E-02

Y( 1,0)=  1.0462435E 00,      ERROR BOUND=  7.4505806E-08

     COEFFICIENTS FOR X BETWEEN X(   0) AND X(   1)

   DELTA=   9.9999995E-01

Y( 1,0)=  1.0000000E 00
Y( 1,1)=  9.9999998E-01
Y( 1,2)=  9.9999994E-01
Y( 1,3)=  9.9999989E-01
Y( 1,4)=  9.9999987E-01
Y( 1,5)=  9.9999983E-01
Y( 1,6)=  9.9999980E-01
Y( 1,7)=  9.9999976E-01
Y( 1,8)=  9.9999974E-01
Y( 1,9)=  1.2857740E 00


SOLUTION AT X(   2)=  1.7553912E-01

Y( 1,0)=  1.2129139E 00,      ERROR BOUND=  1.1920929E-07

     COEFFICIENTS FOR X BETWEEN X(   1) AND X(   2)

   DELTA=   9.1355442E-01

Y( 1,0)=  1.0462435E 00
Y( 1,1)=  9.9999995E-01
Y( 1,2)=  9.5580035E-01
Y( 1,3)=  9.1355435E-01
Y( 1,4)=  8.7317560E-01
Y( 1,5)=  8.3458159E-01
Y( 1,6)=  7.9769341E-01
Y( 1,7)=  7.6243567E-01
Y( 1,8)=  7.2873632E-01
Y( 1,9)=  1.8754066E 00
```

SOLUTION AT X(25)= 9.2183263E-01

Y(1,0)= 1.2793067E 01, ERROR BOUND= 1.5974045E-05

COEFFICIENTS FOR X BETWEEN X(24) AND X(25)

DELTA= 7.5048791E-03

Y(1,0)= 1.1543238E 01
Y(1,1)= 9.9999781E-01
Y(1,2)= 8.6630421E-02
Y(1,3)= 7.5048463E-03
Y(1,4)= 6.5014942E-04
Y(1,5)= 5.6322842E-05
Y(1,6)= 4.8792822E-06
Y(1,7)= 4.2269520E-07
Y(1,8)= 3.6618344E-08
Y(1,9)= 6.0203208E-09

SOLUTION AT X(46)= 9.9097500E-01

Y(1,0)= 1.1080359E 02, ERROR BOUND= 1.2378693E-03

COEFFICIENTS FOR X BETWEEN X(45) AND X(46)

DELTA= 1.0004043E-04

Y(1,0)= 9.9978788E 01
Y(1,1)= 9.9997989E-01
Y(1,2)= 1.0001719E-02
Y(1,3)= 1.0003640E-04
Y(1,4)= 1.0005562E-06
Y(1,5)= 1.0007483E-08
Y(1,6)= 1.0009405E-10
Y(1,7)= 1.0011328E-12
Y(1,8)= 1.0013251E-14
Y(1,9)= 1.9007364E-16

```
SOLUTION AT X( 87)=  9.9986639E-01

Y( 1,0)=  7.4860633E 03,      ERROR BOUND=  5.6754150E 00

     COEFFICIENTS FOR X BETWEEN X( 86) AND X( 87)

   DELTA=   2.1880206E-08

Y( 1,0)=  6.7558077E 03
Y( 1,1)=  9.9863358E-01
Y( 1,2)=  1.4761662E-04
Y( 1,3)=  2.1820492E-08
Y( 1,4)=  3.2254777E-12
Y( 1,5)=  4.7678527E-16
Y( 1,6)=  7.0478013E-20
Y( 1,7)=  1.0417985E-23
Y( 1,8)=  1.5399764E-27
Y( 1,9)=  4.3312046E-31


PROGRAM UNABLE TO BOUND DERIVATIVES OVER-

X =  9.9986639E-01

Y( 1,0)=  7.4860641E 03,      ERROR BOUND=  5.6762085E 00

TIME REQUIRED FOR COMPUTATION=    0.50  MINUTES
```

Example 2 concerns the sin/cos equations, (see pp.14-15). The two cases presented illustrate output options, first of pringtin at every DELTAX=.5 up to XFINAL=9.0 and secondly at the table of values indicated at the bottom of page 15. Notice that the Taylor coefficients are not given.

A third run was made in which we allowed every computed point to be printed along with the Taylor coefficients. We have not included those results. We can summarize them by stating that the program required 17 points to reach a value of X greater than 9.0. The average step size, i.e., increment in X betwen computed points was about 0.54. The computation took 0.11 minutes.

```
    SIN/COS EQUATIONS 1/15/64
INPUT DATA
XZERO=  0.                   XFINAL=  9.0000000E 00     MAXIMUM NO. POINTS=32767
Y( 1,0)=  0.           ,       ERROR BOUND=  C.
Y( 2,0)=  1.0000000E 00,       ERROR BOUND=  7.4505806E-09


SOLUTION AT X=   5.0000000E-01
Y( 1,0)=  4.7942554E-01,     ERROR BOUND=  1.8626451E-08
Y( 2,0)=  8.7758256E-01,     ERROR BOUND=  2.2351742E-08


SOLUTION AT X=   1.0000000E 0C
Y( 1,0)=  8.4147098E-01,     ERROR BOUND=  4.8428774E-08
Y( 2,0)=  5.4030229E-01,     ERROR BOUND=  5.5879354E-08
```

SOLUTION AT X= 1.5000000E 00

Y(1,0)= 9.97749498E-01, ERROR BOUND= 8.9406967E-08

Y(2,0)= 7.07737193E-02, ERROR BOUND= 9.5460564E-08

SOLUTION AT X= 2.0000000E 00

Y(1,0)= 9.09297410E-01, ERROR BOUND= 1.6763806E-07

Y(2,0)= -4.16146840E-01, ERROR BOUND= 1.7508864E-07

SOLUTION AT X= 2.5000000E 00

Y(1,0)= 5.98472120E-01, ERROR BOUND= 2.9429793E-07

Y(2,0)= -8.01143600E-01, ERROR BOUND= 3.0174851E-07

SOLUTION AT X= 3.0000000E 00

Y(1,0)= 1.41120000E-01, ERROR BOUND= 4.9732625E-07

Y(2,0)= -9.89992480E-01, ERROR BOUND= 4.9918890E-07

SOLUTION AT X= 3.5000000E 00

Y(1,0)= -3.50783220E-01, ERROR BOUND= 8.3260238E-07

Y(2,0)= -9.36456670E-01, ERROR BOUND= 8.3446503E-07

SOLUTION AT X= 4.0000000E 00

Y(1,0)= -7.5680248E-01, ERROR BOUND= 1.3932586E-06

Y(2,0)= -6.5364360E-01, ERROR BOUND= 1.3858080E-06

SOLUTION AT X= 4.5000000E 00

Y(1,0)= -9.7753009E-01, ERROR BOUND= 2.3096800E-06

Y(2,0)= -2.1079578E-01, ERROR BOUND= 2.3115426E-06

SOLUTION AT X= 5.0000000E 00

Y(1,0)= -9.5892423E-01, ERROR BOUND= 3.8221478E-06

Y(2,0)= 2.8366219E-01, ERROR BOUND= 3.8221478E-06

SOLUTION AT X= 5.5000000E 00

Y(1,0)= -7.0554028E-01, ERROR BOUND= 6.3218176E-06

Y(2,0)= 7.0866975E-01, ERROR BOUND= 6.3218176E-06

SOLUTION AT X= 6.0000000E 00

Y(1,0)= -2.7941547E-01, ERROR BOUND= 1.0436401E-05

Y(2,0)= 9.6017025E-01, ERROR BOUND= 1.0438263E-05

```
SOLUTION AT X=   6.5000000E 00
Y( 1,0)=  2.1512000E-01,      ERROR BOUND=  1.7217360E-05
Y( 2,0)=  9.7658757E-01,      ERROR BOUND=  1.7218292E-05


SOLUTION AT X=   7.0000000E 00
Y( 1,0)=  6.5698658E-01,      ERROR BOUND=  2.8412789E-05
Y( 2,0)=  7.5390220E-01,      ERROR BOUND=  2.8405339E-05


SOLUTION AT X=   7.5000000E 00
Y( 1,0)=  9.3799993E-01,      ERROR BOUND=  4.6852976E-05
Y( 2,0)=  3.4663527E-01,      ERROR BOUND=  4.6849251E-05


SOLUTION AT X=   8.0000000E 00
Y( 1,0)=  9.8935819E-01,      ERROR BOUND=  7.7251345E-05
Y( 2,0)= -1.4550005E-01,      ERROR BOUND=  7.7255070E-05


SOLUTION AT X=   8.5000000E 00
Y( 1,0)=  7.9848705E-01,      ERROR BOUND=  1.2738258E-04
Y( 2,0)= -6.0201188E-01,      ERROR BOUND=  1.2738630E-04
```

```
SOLUTION AT X=   9.0000000E 00
Y( 1,0)=  4.1211843E-01,      ERROR BOUND=  2.1003745E-04
Y( 2,0)= -9.1113021E-01,      ERROR BOUND=  2.1003932E-04
TIME REQUIRED FOR COMPUTATION=    0.05  MINUTES


SIN/COS EQUATIONS CASE 2 1/15/64
INPUT DATA
XZERO=  0.                 XFINAL=  1.5700000E 00     MAXIMUM NO. POINTS=32767
Y( 1,0)=  0.            ,      ERROR BOUND=  0.
Y( 2,0)=  1.0000000E 00,       ERROR BOUND=  7.4505806E-09

SOLUTION AT X=   1.2300000E-01
Y( 1,0)=  1.2269009E-01,      ERROR BOUND=  3.2596290E-09
Y( 2,0)=  9.9244503E-01,      ERROR BOUND=  1.1175871E-08

SOLUTION AT X=   2.3400000E-01
Y( 1,0)=  2.3187036E-01,      ERROR BOUND=  6.5192580E-09
Y( 2,0)=  9.7274669E-01,      ERROR BOUND=  1.1175871E-08

SOLUTION AT X=   3.4500000E-01
Y( 1,0)=  3.3819668E-01,      ERROR BOUND=  1.3038516E-08
Y( 2,0)=  9.4107546E-01,      ERROR BOUND=  2.2351742E-08
```

```
SOLUTION AT X=    4.5599999E-01
Y( 1,0)=   4.4036036E-01,        ERROR BOUND=   1.4901161E-08
Y( 2,0)=   8.9782112E-01,        ERROR BOUND=   2.2351742E-08

SOLUTION AT X=    5.6699999E-01
Y( 1,0)=   5.3710392E-01,        ERROR BOUND=   2.6077032E-08
Y( 2,0)=   8.4351608E-01,        ERROR BOUND=   2.6077032E-08

SOLUTION AT X=    1.0000000E 00
Y( 1,0)=   8.4147098E-01,        ERROR BOUND=   4.8428774E-08
Y( 2,0)=   5.4030229E-01,        ERROR BOUND=   5.5879354E-08

SOLUTION AT X=    1.5700000E 00
Y( 1,0)=   9.9999968E-01,        ERROR BOUND=   9.6857548E-08
Y( 2,0)=   7.9632643E-04,        ERROR BOUND=   1.0523945E-07
TIME REQUIRED FOR COMPUTATION=    0.01  MINUTES
```

The equations given as example 3 on page 16 are those of the "restricted three-body problem". The value we have picked for μ or C1 makes these equations a mathematical model of the Earth-Moon system in which the solution for Y1, Y2, Y3, Y4 describes the position and velocity components of a small third body, e.g., a lunar spaceship, in "free flight" starting near the surface of the Earth and reaching an altitude of about 13,400 miles at the last computed point. We include only part of the output, namely the 1st, 2nd, 30th, 31st, 55th, and 56th points. Notice the special message printed at the end of the run by the computer showing how to continue the solution.

```
   1/27/64     3-BODY PROBLEM
 INPUT DATA
 XZERO=  0.                  XFINAL=  1.7014100E 38      MAXIMUM NO. POINTS=32767
 Y( 1,C)= -1.9801200E-02,      ERROR BOUND=  1.1641532E-10
 Y( 2,C)= -1.5016200E-02,      ERROR BOUND=  5.8207661E-11
 Y( 3,C)=  9.5560068E 00,      ERROR BOUND=  5.9604645E-08
 Y( 4,C)= -4.8568780E 00,      ERROR BOUND=  2.9802322E-08
 C( 1)=  1.2150000E-02,      ERROR BOUND=  5.8207661E-11
 C( 2)   1.0000000E 00,      ERROR BOUND=  7.4505806E-09
 C( 3)=  2.0000000E 00,      ERROR BOUND=  1.4901161E-08
 FOR INTERMEDIATE VALUES, IF X IS BETWEEN X(Q) AND X(Q+1), SET X = X(Q) + T*DELTA
                  (WHERE T*DELTA IS T MULTIPLIED BY DELTA)
    DETERMINE T AND COMPUTE Y(P,0) AT (X(Q)+T*DELTA) = SUM OVER J OF (T TO THE POWER J TIMES Y(P,J))
    THIS VALUE WILL BE AT LEAST AS ACCURATE AS Y(P,0) AT X(Q+1).
```

```
SOLUTION AT X(   1)=   7.6371821E-05

Y( 1,0)= -1.9066961E-02,      ERROR BOUND=  1.1641532E-09

Y( 2,0)= -1.5378076E-02,      ERROR BOUND=  5.8207661E-10

Y( 3,0)=  9.6700418E 00,      ERROR BOUND=  5.9604645E-07

Y( 4,0)= -4.6189236E 00,      ERROR BOUND=  3.2782555E-07

     COEFFICIENTS FOR X BETWEEN X(   0) AND X(   1)

   DELTA=   3.2469636E-04

Y( 1,0)= -1.9801200E-02
Y( 1,1)=  3.1028007E-03
Y( 1,2)=  8.2722484E-05
Y( 1,3)= -1.1198839E-05
Y( 1,4)= -5.9016575E-07
Y( 1,5)=  1.1815480E-07
Y( 1,6)=  7.3164756E-09
Y( 1,7)= -1.6601855E-09
Y( 1,8)= -1.1098340E-10
Y( 1,9)=  4.7531084E-11

Y( 2,0)= -1.5016200E-02
Y( 2,1)= -1.5770106E-03
Y( 2,2)=  1.6234816E-04
Y( 2,3)=  5.7348511E-06
Y( 2,4)= -1.1495333E-06
Y( 2,5)= -6.0963887E-08
Y( 2,6)=  1.4144252E-08
Y( 2,7)=  8.6305473E-10
Y( 2,8)= -2.1295118E-10
Y( 2,9)= -9.6000516E-12

Y( 3,0)=  9.5560068E 00
Y( 3,1)=  5.0953748E-01
Y( 3,2)= -1.0347057E-01
Y( 3,3)= -7.2703709E-03
Y( 3,4)=  1.8194660E-03
Y( 3,5)=  1.3519971E-04
Y( 3,6)= -3.5791278E-05
Y( 3,7)= -2.7344539E-06
Y( 3,8)=  7.4071255E-07
Y( 3,9)=  4.9471424E-08

Y( 4,0)= -4.8568780E 00
Y( 4,1)=  9.9999988E-01
Y( 4,2)=  5.2986590E-02
Y( 4,3)= -1.4161332E-02
Y( 4,4)= -9.3878301E-04
Y( 4,5)=  2.6136884E-04
Y( 4,6)=  1.8606254E-05
Y( 4,7)= -5.2467771E-06
Y( 4,8)= -3.8795230E-07
Y( 4,9)=  2.2740258E-07
```

SOLUTION AT X(2)= 2.2234346E-04

Y(1,0)= -1.7641402E-02, ERROR BOUND= 1.3969839E-09

Y(2,0)= -1.6018386E-02, ERROR BOUND= 1.1641532E-09

Y(3,0)= 9.8544878E 00, ERROR BOUND= 7.1525574E-07

Y(4,0)= -4.1519712E 00, ERROR BOUND= 4.4703484E-07

COEFFICIENTS FOR X BETWEEN X(1) AND X(2)

DELTA= 3.1754189E-04

Y(1,0)= -1.9066961E-02
Y(1,1)= 3.0706434E-03
Y(1,2)= 7.1386980E-05
Y(1,3)= -1.0931352E-05
Y(1,4)= -4.0789376E-07
Y(1,5)= 1.1314722E-07
Y(1,6)= 3.8855979E-09
Y(1,7)= -1.5514925E-09
Y(1,8)= -4.2503273E-11
Y(1,9)= 6.0577383E-11

Y(2,0)= -1.5378076E-02
Y(2,1)= -1.4667017E-03
Y(2,2)= 1.5877092E-04
Y(2,3)= 4.3244277E-06
Y(2,4)= -1.1060457E-06
Y(2,5)= -3.5926131E-08
Y(2,6)= 1.3317593E-08
Y(2,7)= 3.7667362E-10
Y(2,8)= -1.9519116E-10
Y(2,9)= 3.4794702E-11

Y(3,0)= 9.6700418E 00
Y(3,1)= 4.4962244E-01
Y(3,2)= -1.0327474E-01
Y(3,3)= -5.1381410E-03
Y(3,4)= 1.7816110E-03
Y(3,5)= 7.3418934E-05
Y(3,6)= -3.4201621E-05
Y(3,7)= -1.0708073E-06
Y(3,8)= 6.8724109E-07
Y(3,9)= -1.2124519E-07

Y(4,0)= -4.6189236E 00
Y(4,1)= 9.9999986E-01
Y(4,2)= 4.0855343E-02
Y(4,3)= -1.3932596E-02
Y(4,4)= -5.6569121E-04
Y(4,5)= 2.5163784E-04
Y(4,6)= 8.3035197E-06
Y(4,7)= -4.9175535E-06
Y(4,8)= -1.1720724E-07
Y(4,9)= 3.1836481E-07

SOLUTION AT X(30)= 3.9834206E-03

Y(1,0)= 1.6266221E-02, ERROR BOUND= 1.4901161E-08

Y(2,0)= -1.5465707E-02, ERROR BOUND= 1.6938429E-08

Y(3,0)= 7.2914188E 00, ERROR BOUND= 7.5697899E-06

Y(4,0)= 2.3622525E 00, ERROR BOUND= 7.3164701E-06

COEFFICIENTS FOR X BETWEEN X(29) AND X(30)

DELTA= 1.1661213E-03

Y(1,0)= 1.5015231E-02
Y(1,1)= 8.6692815E-03
Y(1,2)= -5.8305927E-04
Y(1,3)= 5.2393462E-05
Y(1,4)= -1.7069411E-06
Y(1,5)= -1.1371837E-06
Y(1,6)= 4.6950799E-07
Y(1,7)= -1.2532781E-07
Y(1,8)= 2.6247373E-08
Y(1,9)= 4.3660293E-09

Y(2,0)= -1.5860547E-02
Y(2,1)= 2.6631263E-03
Y(2,2)= 3.3212605E-04
Y(2,3)= -8.5546925E-05
Y(2,4)= 1.7244976E-05
Y(2,5)= -2.9798751E-06
Y(2,6)= 4.0212132E-07
Y(2,7)= -2.2970689E-08
Y(2,8)= -9.5260284E-09
Y(2,9)= -1.0903094E-08

Y(3,0)= 7.4342877E 00
Y(3,1)= -9.9999759E-01
Y(3,2)= 1.3478905E-01
Y(3,3)= -5.8551062E-03
Y(3,4)= -4.8759233E-03
Y(3,5)= 2.4157417E-03
Y(3,6)= -7.5231849E-04
Y(3,7)= 1.8006615E-04
Y(3,8)= -3.2095505E-05
Y(3,9)= -5.1439506E-05

Y(4,0)= 2.2837472E 00
Y(4,1)= 5.6962520E-01
Y(4,2)= -2.2008068E-01
Y(4,3)= 5.9153281E-02
Y(4,4)= -1.2776865E-02
Y(4,5)= 2.0690196E-03
Y(4,6)= -1.3788859E-04
Y(4,7)= -6.5351884E-05
Y(4,8)= 3.7177252E-05
Y(4,9)= 6.2712352E-05

```
SOLUTION AT X( 31)=  4.1590815E-03

Y( 1,0)=  1.7534496E-02,     ERROR BOUND=  1.6414560E-08

Y( 2,0)= -1.5044268E-02,     ERROR BOUND=  1.8277206E-08

Y( 3,0)=  7.1495931E 00,     ERROR BOUND=  8.0764294E-06

Y( 4,0)=  2.4346721E 00,     ERROR BOUND=  7.6293945E-06

     COEFFICIENTS FOR X BETWEEN X( 30) AND X( 31)

   DELTA=   1.2132593E-03

Y( 1,0)=  1.6266221E-02
Y( 1,1)=  8.8463817E-03
Y( 1,2)= -6.0662813E-04
Y( 1,3)=  5.7645639E-05
Y( 1,4)= -2.8106945E-06
Y( 1,5)= -9.4881327E-07
Y( 1,6)=  4.5184824E-07
Y( 1,7)= -1.2903889E-07
Y( 1,8)=  2.9019271E-08
Y( 1,9)=  5.6922431E-09

Y( 2,0)= -1.5465707E-02
Y( 2,1)=  2.8660248E-03
Y( 2,2)=  3.2132116E-04
Y( 2,3)= -8.5711106E-05
Y( 2,4)=  1.7809955E-05
Y( 2,5)= -3.2184441E-06
Y( 2,6)=  4.7458050E-07
Y( 2,7)= -4.0751166E-08
Y( 2,8)= -6.0739245E-09
Y( 2,9)= -1.3805263E-08

Y( 3,0)=  7.2914188E 00
Y( 3,1)= -9.9999750E-01
Y( 3,2)=  1.4253912E-01
Y( 3,3)= -9.2665912E-03
Y( 3,4)= -3.9101834E-03
Y( 3,5)=  2.2345507E-03
Y( 3,6)= -7.4450057E-04
Y( 3,7)=  1.9134753E-04
Y( 3,8)= -3.8343199E-05
Y( 3,9)= -5.9504294E-05

Y( 4,0)=  2.3622525E 00
Y( 4,1)=  5.2968258E-01
Y( 4,2)= -2.1193600E-01
Y( 4,3)=  5.8717721E-02
Y( 4,4)= -1.3263628E-02
Y( 4,5)=  2.3469699E-03
Y( 4,6)= -2.3511723E-04
Y( 4,7)= -4.0050297E-05
Y( 4,8)=  3.2568845E-05
Y( 4,9)=  7.0957816E-05
```

SOLUTION AT X(55)= 1.1396442E-02

Y(1,0)= 5.6302577E-02, ERROR BOUND= 1.4924444E-07

Y(2,0)= 5.7992573E-03, ERROR BOUND= 1.0136864E-07

Y(3,0)= 4.2239751E 00, ERROR BOUND= 2.7000904E-05

Y(4,0)= 2.9321976E 00, ERROR BOUND= 1.3872981E-05

COEFFICIENTS FOR X BETWEEN X(54) AND X(55)

DELTA= 4.6183784E-03

Y(1,0)= 5.4266144E-02
Y(1,1)= 1.9968908E-02
Y(1,2)= -2.3091688E-03
Y(1,3)= 5.0380888E-04
Y(1,4)= -1.2841514E-04
Y(1,5)= 3.4808263E-05
Y(1,6)= -9.5619999E-06
Y(1,7)= 2.5651524E-06
Y(1,8)= -6.4067957E-07
Y(1,9)= 9.5953691E-07

Y(2,0)= 4.3990668E-03
Y(2,1)= 1.3596577E-02
Y(2,2)= -2.4933268E-04
Y(2,3)= -1.0558745E-04
Y(2,4)= 5.9510480E-05
Y(2,5)= -2.5400825E-05
Y(2,6)= 1.0075071E-05
Y(2,7)= -3.8652956E-06
Y(2,8)= 1.4512523E-06
Y(2,9)= -9.1843753E-07

Y(3,0)= 4.3237922E 00
Y(3,1)= -9.9999115E-01
Y(3,2)= 3.2726349E-01
Y(3,3)= -1.1122098E-01
Y(3,4)= 3.7684508E-02
Y(3,5)= -1.2422542E-02
Y(3,6)= 3.8879592E-03
Y(3,7)= -1.1097914E-03
Y(3,8)= 2.6150950E-04
Y(3,9)= -1.2887865E-03

Y(4,0)= 2.9440155E 00
Y(4,1)= -1.0797412E-01
Y(4,2)= -6.8587351E-02
Y(4,3)= 5.1542316E-02
Y(4,4)= -2.7499723E-02
Y(4,5)= 1.3089103E-02
Y(4,6)= -5.8585649E-03
Y(4,7)= 2.5138734E-03
Y(4,8)= -1.0427660E-03
Y(4,9)= 8.1849331E-04

```
SOLUTION AT X( 56)=  1.1894594E-02

Y( 1,0)=  5.8382172E-02,     ERROR BOUND=  1.6321428E-07

Y( 2,0)=  7.2565989E-03,     ERROR BOUND=  1.0835356E-07

Y( 3,0)=  4.1263477E 00,     ERROR BOUND=  2.8222799E-05

Y( 4,0)=  2.9186078E 00,     ERROR BOUND=  1.4171004E-05

     COEFFICIENTS FOR X BETWEEN X( 55) AND X( 56)

   DELTA=   4.9349204E-03

Y( 1,0)=  5.6302577E-02
Y( 1,1)=  2.0844980E-02
Y( 1,2)= -2.4674368E-03
Y( 1,3)=  5.5427710E-04
Y( 1,4)= -1.4586612E-04
Y( 1,5)=  4.0988331E-05
Y( 1,6)= -1.1741450E-05
Y( 1,7)=  3.3172033E-06
Y( 1,8)= -8.9029517E-07
Y( 1,9)=  1.2558048E-06

Y( 2,0)=  5.7992573E-03
Y( 2,1)=  1.4470162E-02
Y( 2,2)= -3.1796065E-04
Y( 2,3)= -1.0190057E-04
Y( 2,4)=  6.2414846E-05
Y( 2,5)= -2.7783990E-05
Y( 2,6)=  1.1417753E-05
Y( 2,7)= -4.5317307E-06
Y( 2,8)=  1.7605739E-06
Y( 2,9)= -1.1204358E-06

Y( 3,0)=  4.2239751E 00
Y( 3,1)= -9.9999052E-01
Y( 3,2)=  3.3695199E-01
Y( 3,3)= -1.1823179E-01
Y( 3,4)=  4.1528867E-02
Y( 3,5)= -1.4275550E-02
Y( 3,6)=  4.7053288E-03
Y( 3,7)= -1.4432576E-03
Y( 3,8)=  3.8650850E-04
Y( 3,9)= -1.5346296E-03

Y( 4,0)=  2.9321976E 00
Y( 4,1)= -1.2886151E-01
Y( 4,2)= -6.1946631E-02
Y( 4,3)=  5.0590357E-02
Y( 4,4)= -2.8150393E-02
Y( 4,5)=  1.3881990E-02
Y( 4,6)= -6.4280904E-03
Y( 4,7)=  2.8540665E-03
Y( 4,8)= -1.2262760E-03
Y( 4,9)=  9.5137954E-04
```

```
SOLUTION AT X=  1.2415448E-02
Y( 1,0)=  6.0506257E-02,      ERROR BOUND=  1.7834827E-07
Y( 2,0)=  8.7728341E-03,      ERROR BOUND=  1.1606608E-07
Y( 3,0)=  4.0308846E 00,      ERROR BOUND=  2.9504299E-05
Y( 4,0)=  2.9033469E 00,      ERROR BOUND=  1.4498830E-05
TIME REQUIRED FOR COMPUTATION=    5.29  MINUTES
ELAPSED TIME HAS EXCEEDED THAT ALLOWED BY ESTIMATED RUN TIME.  THE LAST VALUE OF THE INDEPENDENT
VARIABLE  FOR WHICH THE PROGRAM COMPLETED THE SOLUTION, TOGETHER WITH THE SOLUTION VALUES, IS SHOWN
ABOVE.  SHOULD THE USER DESIRE TO CONTINUE INTEGRATION BEYOND THIS POINT, HE MAY USE THE ABOVE RESULTS
AS INPUT DATA FOR THE INITIAL VALUES OF THE INDEPENDENT AND DEPENDENT VARIABLES, AND RE-SUBMIT THE PROGRAM.
```

ACKNOWLEDGMENTS

The authors are indebted to Ray Boche, Hal Robison, Tony Steigler, and others for assistance and advice on various parts of the program. We are indebted to Julie Van Duyn for her care in the typing of the manuscript.

REFERENCES

1. Moore, R. E., "Interval Arithmetic and Automatic Error Analysis in Digital Computing," Technical Report No. 25, 1962, Applied Mathematical and Statistical Laboratory, Stanford University, Stanford, California.

2. Halpern, Mark, "A Programming System for Command and Control Application," LMSC Technical Report, 1963, Lockheed Missiles & Space Company, Palo Alto, California.