GIT/EES-8-11129-64

Gy

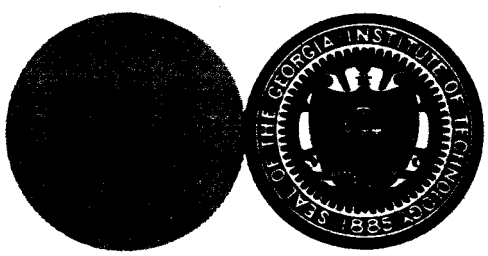
PROJECT A-740

THE APPLICATION OF A NUMERICAL INTEGRATION PROCEDURE DEVELOPED BY ERWIN FEHLBERG TO THE RESTRICTED PROBLEM OF THREE BODIES

I. E. PERLIN AND C. P. REED, JR.

Contract No. NAS8-11129

Prepared for
George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama



Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia

GEORGIA INSTITUTE OF TECHNOLOGY
Engineering Experiment Station
Atlanta, Georgia

PROJECT A-740

THE APPLICATION OF A NUMERICAL INTEGRATION PROCEDURE DEVELOPED BY ERWIN FEHLBERG TO THE RESTRICTED PROBLEM OF THREE BODIES

By

I. E. PERLIN and C. P. REED, JR.

CONTRACT NO. NAS8-11129

Performed for
GEORGE C. MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HUNTSVILLE, ALABAMA

TABLE OF CONTENTS

THE APPLICATION OF A NUMERICAL INTEGRATION PROCEDURE DEVELOPED BY ERWIN FEHLBERG TO THE RESTRICTED PROBLEM OF THREE BODIES

I. The Numerical Method

In a rotating coordinate system, the equations of motion of a space vehicle in the Earth-Moon system are[1]:

$$\ddot{x} = f(x,y,\dot{y}) = x + 2\dot{y} - \mu' \cdot \frac{x+\mu}{[(x+\mu)^2+y^2]^{3/2}} - \mu \cdot \frac{x-\mu'}{[(x-\mu')^2+y^2]^{3/2}} \quad ,$$

$$\ddot{y} = g(x,y,\dot{x}) = y - 2\dot{x} - \mu' \cdot \frac{y}{[(x+\mu)^2+y^2]^{3/2}} - \mu \cdot \frac{y}{[(x-\mu')^2+y^2]^{3/2}} \quad , \qquad (1)$$

where $\mu' = 1-\mu$ with μ representing the relative mass of the Moon.

Introducing into (1) the following four auxiliary functions:

$$r^2 = (x+\mu)^2+y^2 \quad , \quad s^2 = (x-\mu')^2+y^2 \quad ,$$

$$u = \mu'/r^3 \quad , \quad v = \mu/s^3 \quad , \qquad (2)$$

system (1) becomes

$$\ddot{x} = x + 2\dot{y} - u(x+\mu) - v(x-\mu') \quad ,$$

$$\ddot{y} = y - 2\dot{x} - uy - vy \quad ,$$

$$r\dot{u} + 3u\dot{r} = 0 \quad ,$$

$$s\dot{v} + 3v\dot{s} = 0 \quad . \qquad (3)$$

---

[1] Arenstorf, R. F., American Journal of Mathematics, LXXXV, 27 (1963).

Substituting the following power series expansions, in terms of $h_j \equiv (t - t_j)$ about the point t_j, into (3)

$$x = \sum_{i=0} X_i \cdot (h_j)^i \quad , \quad y = \sum_{i=0} Y_i \cdot (h_j)^i \quad ,$$

$$u = \sum_{i=0} U_i \cdot (h_j)^i \quad , \quad v = \sum_{i=0} V_i \cdot (h_j)^i \quad ,$$

$$r = \sum_{i=0} R_i \cdot (h_j)^i \quad , \quad s = \sum_{i=0} S_i \cdot (h_j)^i \quad , \tag{4}$$

and equating the coefficients of like terms, the following recurrence formulas for the power series coefficients in (4) result:

$$R_n = \left\{ \sum_{i=0}^{n} X_i X_{n-i} + 2\mu X_n + \sum_{i=0}^{n} Y_i Y_{n-i} - \sum_{i=1}^{n-1} R_i R_{n-i} \right\} /2R_0 \quad ,$$

$$S_n = \left\{ \sum_{i=0}^{n} X_i X_{n-i} - 2\mu' X_n + \sum_{i=0}^{n} Y_i Y_{n-i} - \sum_{i=1}^{n-1} S_i S_{n-i} \right\} /2S_0 \quad ,$$

$$U_n = \left\{ -3 \sum_{i=1}^{n} i R_i U_{n-i} - \sum_{i=1}^{n-1} i U_i R_{n-i} \right\} /nR_0 \quad ,$$

$$V_n = \left\{ -3 \sum_{i=1}^{n} i S_i V_{n-i} - \sum_{i=1}^{n-1} i V_i S_{n-i} \right\} /nS_0 \quad ,$$

$$X_{n+1} = \left\{ X_{n-1} + 2nY_n - \mu U_{n-1} + \mu' V_{n-1} - \sum_{i=0}^{n-1} (U_i - V_i) X_{n-1-i} \right\} /n(n+1) \quad ,$$

$$Y_{n+1} = \left\{ Y_{n-1} - 2nX_n - \sum_{i=0}^{n-1} (U_i - V_i) Y_{n-1-i} \right\} /n(n+1) \quad , \tag{5}$$

where $n = 1,2,3 \ldots$. For $n = 1$, the last sum of the first four equations in (5) must be omitted.

At the start of the j^{th} integration step, the initial values for the step determine the initial coefficients as follows:

$$X_0 = x_j \quad , \quad Y_0 = y_j \quad ,$$

$$X_1 = \dot{x}_j \quad , \quad Y_1 = \dot{y}_j \quad ,$$

$$R_0 = [(x_j+\mu)^2 + y_j^2]^{1/2} \quad , \quad S_0 = [(x_j-\mu')^2 + y_j^2]^{1/2} \quad ,$$

$$U_0 = \mu'/R_0^3 \quad , \quad V_0 = \mu/S_0^3 \quad . \tag{6}$$

After computing, through formulas (5), a sufficiently large number (m) of consecutive derivatives of x and y with respect to t, in the form of the coefficients X_n and Y_n, we then apply the Runge-Kutta procedure, described in E. Fehlberg's paper[2], to obtain (m+4)-th-order accuracy.

Introducing new variables x_T, y_T, $\dot{x}_T$, and $\dot{y}_T$ by the following transformations

$$x_T = x - \sum_{i=1}^{m+2} X_i (h_j)^i \quad , \quad y_T = y - \sum_{i=1}^{m+2} Y_i (h_j)^i \quad ,$$

$$\dot{x}_T = \dot{x} - \sum_{i=1}^{m+2} iX_i (h_j)^{i-1} \quad , \quad \dot{y}_T = \dot{y} - \sum_{i=1}^{m+2} iY_i (h_j)^{i-1} \quad , \tag{7}$$

the original differential equations (1) are transformed accordingly:

---

[2]Fehlberg, E., Runge-Kutta Type Formulas of High-Order Accuracy and Their Application to the Numerical Integration of the Restricted Problem of Three Bodies. Presented at the "Colloque International des Techniques de Calcul Analogique et Numerique en Aeronautique" in Liege, Belgium, September, 1963, and to be published in the Proceedings of this Symposium.

$$\ddot{x}_T = f_T(t, x_T, y_T, \dot{y}_T) = f(x, y, \dot{y}) - \sum_{i=2}^{m+2} i(i-1)\, X_i\, (h_j)^{i-2} \quad ,$$

$$\ddot{y}_T = g_T(t, x_T, y_T, \dot{x}_T) = g(x, y, x) - \sum_{i=2}^{m+2} i(i-1)\, Y_i\, (h_j)^{i-2} \qquad (8)$$

The initial values of the transformed variables for starting an orbit are

$$(x_T)_0 = x_0 \quad , \quad (y_T)_0 = y_0 \quad ,$$

$$(\dot{x}_T)_0 = 0 \quad , \quad (\dot{y}_T)_0 = 0 \quad . \qquad (9)$$

For equations (8) the following Runge-Kutta formulas develop (m+4)-th-order accuracy for $x, y, \dot{x}$, and $\dot{y}$ with only three substitutions per time step, $\Delta t_j = (t_{j+1} - t_j)$:

$$k_1 = \Delta t_j f_T(t_j + \alpha_1 \Delta t_j \, , \, (x_T)_j \, , \, (y_T)_j \, , \, 0)$$

$$\ell_1 = \Delta t_j g_T(t_j + \alpha_1 \Delta t_j \, , \, (x_T)_j \, , \, (y_T)_j \, , \, 0)$$

$$k_2 = \Delta t_j f_T(t_j + \alpha_2 \Delta t_j \, , \, (x_T)_j + \beta_0 k_1 \Delta t_j \, , \, (y_T)_j + \beta_0 \ell_1 \Delta t_j, \, \beta_1 \ell_1)$$

$$\ell_2 = \Delta t_j g_T(t_j + \alpha_2 \Delta t_j \, , \, (x_T)_j + \beta_0 k_1 \Delta t_j \, , \, (y_T)_j + \beta_0 \ell_1 \Delta t_j, \, \beta_1 k_1)$$

$$k_3 = \Delta t_j f_T(t_j + \alpha_3 \Delta t_j \, , \, (x_T)_j + \gamma_0 k_1 \Delta t_j + \delta_0 k_2 \Delta t_j \, , \, (y_T)_j$$

$$+ \gamma_0 \ell_1 \Delta t_j + \delta_0 \ell_2 \Delta t_j \, , \, \gamma_1 \ell_1 + \delta_1 \ell_2)$$

$$\ell_3 = \Delta t_j g_T(t_j+\alpha_3\Delta t_j\ ,\ (x_T)_j + \gamma_0 k_1 \Delta t_j + \delta_0 k_2 \Delta t_j\ ,\ (y_T)_j$$

$$+ \gamma_0 \ell_1 \Delta t_j + \delta_0 \ell_2 \Delta t_j\ ,\ \gamma_1 k_1 + \delta_2 k_2) \qquad (10)$$

and

$$(x_T)_{j+1} = (x_T)_j + (C_1k_1+C_2k_2+C_3k_3)\ \Delta t_j$$

$$(y_T)_{j+1} = (y_T)_j + (C_1\ell_1+C_2\ell_2+C_3\ell_3)\ \Delta t_j$$

$$(\dot{x}_T)_{j+1} = C_1'k_1+C_2'k_2+C_3'k_3$$

$$(\dot{y}_T)_{j+1} = C_1'\ell_1+C_2'\ell_2+C_3'\ell_3 \qquad (11)$$

where

$$\alpha_1 = 1\ ,\ \alpha_2 = \frac{m+2}{m+4}\ ,\ \alpha_3 = 1\ ,$$

$$\beta_0 = \frac{2}{(m+4)^2}\cdot\left(\frac{m+2}{m+4}\right)^{m+1},\ \beta_1 = \frac{1}{m+4}\cdot\left(\frac{m+2}{m+4}\right)^{m+1},$$

$$\gamma_0 = 0\ ,\ \gamma_1 = -\frac{1}{m+2}\ ,\ \delta_0 = 0\ ,\ \delta_1 = \frac{2}{m+2}\cdot\left(\frac{m+4}{m+2}\right)^{m+1},$$

$$C_1 = 0\ ,\ C_2 = \frac{1}{(m+2)(m+3)}\cdot\left(\frac{m+4}{m+2}\right)^{m+1},\ C_3 = 0$$

$$C_1' = 0\ ,\ C_2' = \frac{1}{2}\cdot\frac{m+4}{(m+2)(m+3)}\cdot\left(\frac{m+4}{m+2}\right)^{m+1},\ C_3' = \frac{1}{2}\cdot\frac{1}{m+3} \qquad (12)$$

A fourth substitution yields improved values, $\hat{x}_T$ and $\hat{y}_T$, which are more accurate by one power of Δt_j

$$k_4 = \Delta t_j f_T(t_j+\alpha_4\Delta t_j\ ,\ (x_T)_j + \epsilon_0 k_1\Delta t_j + \zeta_0 k_2\Delta t_j + \eta_0 k_3\Delta t_j\ ,$$

$$(y_T)_j + \epsilon_0 \ell_1\Delta t_j + \zeta_0 \ell_2\Delta t_j + \eta_0 \ell_3\Delta t_j\ ,\ \epsilon_1\ell_1 + \zeta_1\ell_2 + \eta_1\ell_3)$$

$$\ell_4 = \Delta t_j g_T(t_j+\alpha_4\Delta t_j\ ,\ (x_T)_j + \epsilon_0 k_1\Delta t_j + \zeta_0 k_2\Delta t_j + \eta_0 k_3\Delta t_j\ ,$$

$$(y_T)_j + \epsilon_0 \ell_1\Delta t_j + \zeta_0 \ell_2\Delta t_j + \eta_0 \ell_3\Delta t_j\ ,\ \epsilon_1 k_1 + \zeta_1 k_2 + \eta_1 k_3) \qquad (13)$$

and

$$(\hat{x}_T)_{j+1} = (x_T)_j + (\hat{C}_1 k_1 + \hat{C}_2 k_2 + \hat{C}_3 k_3 + \hat{C}_4 k_4)\ \Delta t_j$$

$$(\hat{y}_T)_{j+1} = (y_T)_j + (\hat{C}_1 \ell_1 + \hat{C}_2 \ell_2 + \hat{C}_3 \ell_3 + \hat{C}_4 \ell_4)\ \Delta t_j \qquad (14)$$

where

$$\alpha_4 = \frac{m+2}{m+5}$$

$$\epsilon_0 = 0\ ,\ \epsilon_1 = \frac{1}{2} \cdot \left(\frac{1}{m+5}\right) \cdot \left(\frac{m+2}{m+5}\right)^{m+1}$$

$$\zeta_0 = 0\ ,\ \zeta_1 = \frac{1}{4} \cdot \frac{5m+16}{(m+5)^2} \cdot \left(\frac{m+4}{m+5}\right)^{m+1}$$

$$\eta_0 = \frac{3}{2} \cdot \frac{1}{(m+5)^2} \cdot \left(\frac{m+2}{m+5}\right)^{m+1},$$

$$\eta_1 = -\frac{3}{4} \cdot \frac{m+2}{(m+5)^2} \cdot \left(\frac{m+2}{m+5}\right)^{m+1}$$

$$\hat{C}_1 = 0 \ , \ \hat{C}_2 = 0 \ , \ \hat{C}_3 = \frac{1}{3} \ \cdot \ \frac{1}{(m+3)(m+4)}$$

$$\hat{C}_4 = \frac{2}{3} \cdot \frac{m+5}{(m+2)(m+3)(m+4)} \ \cdot \left(\frac{m+5}{m+2}\right)^{m+1} \tag{15}$$

The approximate value of the truncation error in both variables is given by

$$(T_x)_{j+1} \approx \left|(x_T)_{j+1} - (\hat{x}_T)_{j+1}\right|$$

$$(T_y)_{j+1} \approx \left|(y_T)_{j+1} - (\hat{y}_T)_{j+1}\right| \tag{16}$$

Since the evaluation of f_T and g_T, necessary in equations (10) and (13), can be accomplished only through the relationships given in (8), the following inverse transformations must be executed:

$$x_{j+1} = (x_T)_{j+1} + \sum_{i=1}^{m+2} X_i \ (\Delta t_j)^i$$

$$y_{j+1} = (y_T)_{j+1} + \sum_{i=1}^{m+2} Y_i \ (\Delta t_j)^i$$

$$\dot{x}_{j+1} = (\dot{x}_T)_{j+1} + \sum_{i=1}^{m+2} i \ X_i \ (\Delta t_j)^{i-1}$$

$$\dot{y}_{j+1} = (\dot{y}_T)_{j+1} + \sum_{i=1}^{m+2} i \ Y_i \ (\Delta t_j)^{i-1} \tag{17}$$

thus allowing values of $(f)_{j+1}$ and $(g)_{j+1}$ to be obtained through equations (1). Equations (8) now yield values of $(f_T)_{j+1}$ and $(g_T)_{j+1}$.

II. The Computer Program

a) General

A Burroughs B-5000 Algol program was devised to implement the above integration procedure. (See flow diagram, figure 1, and PROGRAM LISTING) Modular construction was emphasized in the design of the program to facilitate numerical investigation of the method.

Given a set of initial conditions, there are three independent options under which the program may be executed.

One option produces values of $x, y, \dot{x}, \dot{y}, t$, the step size, the Jacobi constant, the distance to the Earth, and the distance to the Moon at each time step for a full period (see EXAMPLE ORBIT). A period is determined to be complete when $|y| < \epsilon_1$ after a given time equal to 90% of the approximate time for one period. The value of ϵ_1 and the approximate time for one period are input parameters, and a step-size-halving process is repeated until the specified condition is satisfied.

Another option is included to force an iterative procedure for improving a given set of initial conditions until an orbit is produced which is periodic within a prescribed error tolerance. The technique employed consists of interpolating on successive half-orbit values of $\dot{x}$ and $\dot{y}$ to produce corrected initial values of $\dot{y}$. The process terminates when at half-period, $|\dot{x}| < \epsilon_2$. The value of ϵ_2 is an input parameter and the half-period point is determined in a manner similar to that described for determining the full period point. The user may elect to print each successive set of improved initial conditions, (see IMPROVED INITIAL CONDITIONS), and values for the final orbit are printed automatically as in the previously described option.

Under a third option, an output magnetic tape is prepared with a content and format acceptable to the SC 4020 plotter. In this manner the coordinates

of every computed point describing an orbit may be plotted as well as printed. (See figure 2).

Automatic step size control to regulate the truncation error is an inherent part of the program. Values for a particular time step are accepted if, and only if, neither of the following conditions exist:

$$\frac{T_x}{\epsilon|x|} > 1 \quad , \quad \frac{T_y}{\epsilon|y|} > 1$$

where T_x and T_y are measures of the truncation error as described in Section I, equations (16), and ϵ is an input parameter. If either condition occurs, the step size is halved and the values are recomputed.

Doubling of the step size for the next time step will occur if, and only if, both the following conditions are satisfied for the current time step:

$$\frac{T_x}{\epsilon|x|} < \left(\frac{1}{2}\right)^{m+5} \quad , \quad \frac{T_y}{\epsilon|y|} < \left(\frac{1}{2}\right)^{m+5}$$

where m has the same meaning described in Section I, page 3.

b) Hardware Requirements

The program may be executed on a minimum Burroughs B-5000 system. A card reader and line printer are required under all options and a tape unit is needed if the plotting option is specified.

c) Input Data

Input data for one complete orbit is supplied to the program in sets, hereafter referred to as data-sets, and there is no limit to the number of data-sets that can be queued in the card reader for processing.

A data set consists of either 9 or 11 cards, depending on the choice of the plotting option, and must follow the program cards.

Data Card 1 contains a value for μ, the ratio of the mass of the Moon to the mass of the Earth. It appears in columns 1 through 18 as a number in scientific notation, 11 significant digits, as follows:

$$\pm 0 . ***********@ \pm EE$$

Data Card 2 contains values which are used to stop computation on ill-behaved orbits. The values as they appear from left to right on the card are as follows:

r_{max} - the maximum allowable distance between the orbiting body and the Earth, columns 1 through 15, in scientific notation, 8 significant digits.

r_{min} - the minimum allowable distance between the orbiting body and the Earth, columns 16 through 30, in scientific notation, 8 significant digits.

s_{max} - the maximum allowable distance between the orbiting body and the Moon, columns 31 through 45, in scientific notation, 8 significant digits.

s_{min} - the minimum allowable distance between the orbiting body and the Moon, columns 46 through 60, in scientific notation, 8 significant digits.

t_{max} - the maximum allowable time for one orbit, columns 61 through 75, in scientific notation, 8 significant digits.

Data Card 3 contains a value of t_{start}, the starting time for an orbit, columns 1 through 15, in scientific notation, 8 significant digits.

Data Card 4 contains values of the error tolerances allowable. The values to 8 significant digits as they appear from left to right are as follows:

ϵ - the truncation error tolerance for halving the interval size in the test,

$$\frac{T_x}{\epsilon . |x|} > 1 ,$$

columns 1 through 15, in scientific notation.

ϵ_1 - the error tolerance involved in the test, $|y| < \epsilon_1$, during the iterative procedure for producing periodic orbits, columns 16 through 30, in scientific notation.

ϵ_2 - the error tolerance involved in the test, $|\dot{x}| < \epsilon_2$, during the iterative procedure for producing periodic orbits, columns 31 through 45, in scientific notation.

Data Card 5 contains a value for h_{start}, the initial time step interval, columns 1 through 15, in scientific notation, 8 significant digits.

Data Card 6 contains a value of m, the number of terms of the power series expansion to be considered. It appears in columns 1 through 3 in integral notation as follows:

b**

Data Card 7 contains values of p, the period of the orbit; d_1, the direction of crossing at half period; and d_2, the direction of crossing at full period. d_1 and d_2 equal +1 if the crossing is from plus to minus, and -1 otherwise. A value of d_1 must be supplied but is relevant only under the iterative procedure option. The values appear on the card as follows:

p - columns 1 through 15, in scientific notation, 8 significant digits

d_1 - columns 16, 17, an integer (+1 or -1)

d_2 - columns 18, 19, an integer (+1 or -1)

Data Card 8 contains the initial values of the variables x_o, y_o, $\dot{x}_o$, $\dot{y}_o$. The values appear on the card as follows:

x_o - in columns 1 through 18, in scientific notation, 11 significant digits

y_o - in columns 19 through 36, in scientific notation, 11 significant digits

$\dot{x}_o$ - in columns 37 through 54, in scientific notation, 11 significant digits

$\dot{y}_o$ - in columns 55 through 72, in scientific notation, 11 significant digits

Data Card 9 is used to choose the program options as follows:

If the iteration option is desired, columns 1 through 5 contain bTRUE otherwise FALSE.

If printing of each corrected set of initial values produced during the iteration is desired, columns 6 through 10 contain bTRUE otherwise FALSE. If the iterative option is "false" a value for the print iteration option must be supplied but is irrelevant.

If the plot option is desired, columns 11 through 15 must contain bTRUE otherwise FALSE.

Data Card 10 is required only under the plot option. It contains a value of the current date in columns 1 through 6 as follows:

columns 1 and 2, month, **

columns 3 and 4, day, **

columns 5 and 6, year **

Data Card 11 is required only under the plot option and contains in columns 1 through 48 any desired identification information to be associated with the plot of the orbit.

d) Operating Instructions

There are no special operating instructions. Choice of the plotting option requires the presence of an unlabeled scratch tape on some available unit.

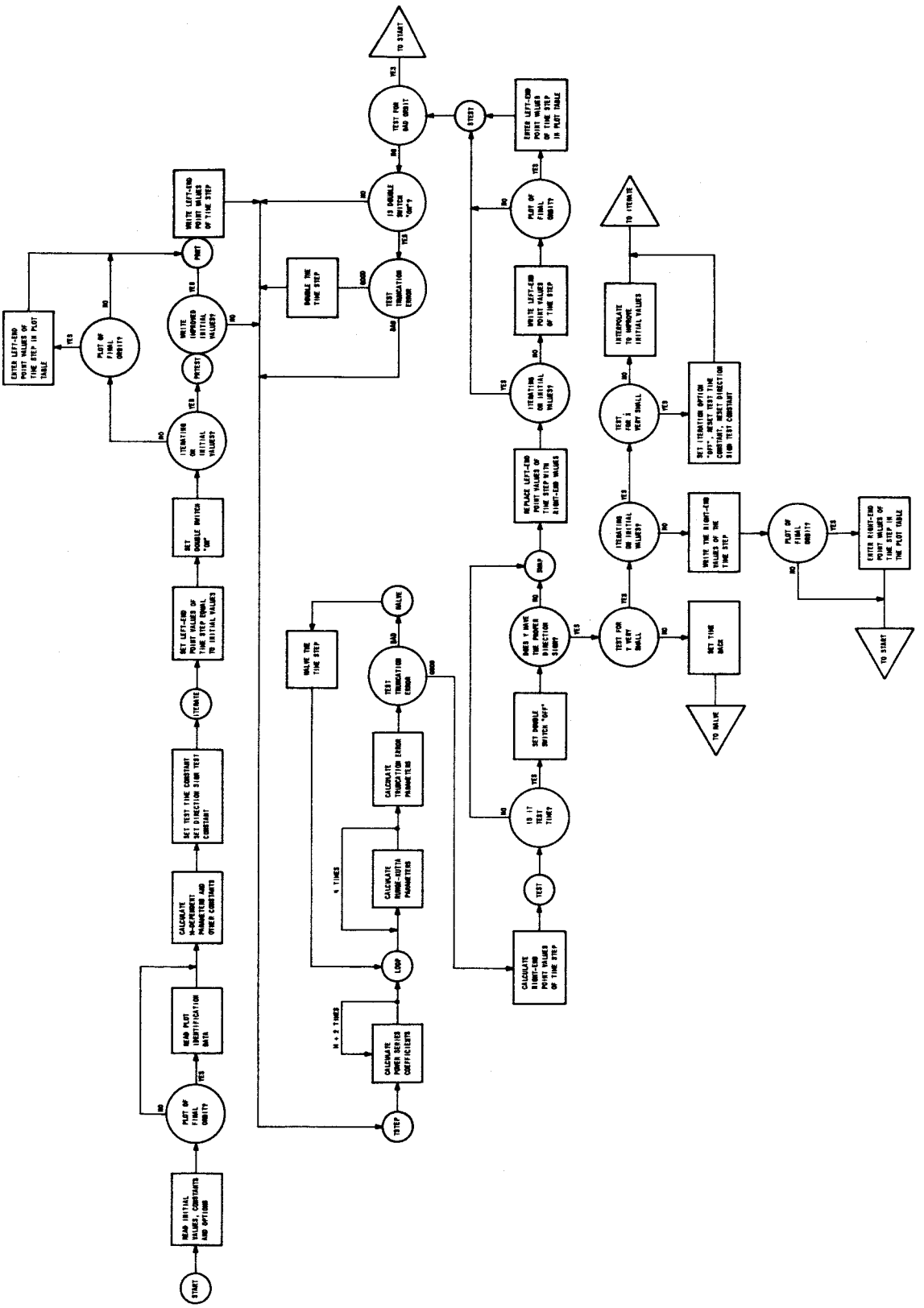


Figure 1. Flow Diagram.

PROGRAM LISTING

The following is a listing of the program deck, including control cards, and the data cards which produced the included example.

Input values for the example are:

$\mu = 0.12128562765 \times 10^{-1}$

$r_{max} = 0.3 \times 10^{1}$

$r_{min} = 0.16574301 \times 10^{-1}$

$s_{max} = 0.3 \times 10^{1}$

$s_{min} = 0.45212967 \times 10^{-2}$

$t_{max} = 0.7 \times 10^{1}$

$t_0 = 0$

$\epsilon = 0.1 \times 10^{-6}$

$\epsilon_1 = 0.1 \times 10^{-8}$

$\epsilon_2 = 0.1 \times 10^{-8}$

$h_0 = 0.125$

$m = 8$

$p = 0.628 \times 10^{1}$

$d_1 = -1$

$d_2 = +1$

$x_0 = 0.12 \times 10^{1}$

$y_0 = 0$

$\dot{x}_0 = 0$

$\dot{y}_0 = - 0.1049 \times 10^{1}$

iter = true

pritr = true

plt = true

```
930                                                                    9
 LABEL  0000000000XXXXXX0010000001
 BEGIN                                                                 001
COMMENT FOLLOWING IS THE SQUEEZED VERSION OF THE SC 4020 PLOT PROGEDURE;
 INTEGER WRITE4020SWITCH; INTEGER GRAPHCTPERPAGE,TOMJIMJOEJOHN,XYZWPAGECSC     1
T; INTEGER ARRAY THISISIDENTQ[0:500], DRATTYA[0:63]; ALPHA ARRAY THISISJSC     2
ORID[1:16],B222IDARRAY[1:8]; FILE OUT FT4020 9(2,500)); PROCEDURE SETSCALSC     3
E(X,Y,TESTLOC); VALUE TESTLOC; COMMENT IF TESTLOC=0 THEN SET NEW X AND YSC     4
 VALUES ELSE RETRIEVE SCALE VALUES; INTEGER X,Y,TESTLOC; BEGIN OWN INTEGSC     5
ER SAVEX,SAVEY; LABEL OUD; IF TESTLOC=0 THEN BEGIN SAVEX ← X; SAVEY ← Y;SC     6
 GO TO OUD END ELSE BEGIN X ← SAVEX; Y ← SAVEY END; OUD: END; REAL PROCESC     7
DURE XMODV(A); REAL A; BEGIN XMODV ← LN(A) END; REAL PROCEDURE YMODV(A);SC     8
 REAL A; BEGIN YMODV ← LN(A) END; PROCEDURE XSCALV(XL,XR,ML,MR,TEST,IM,OSC     9
UD,ERROR); VALUE XL,XR,ML,MR,TEST,IM; INTEGER ML,MR,TEST,OUD,ERROR; REALSC    10
 XL,XR,IM; BEGIN INTEGER LO,HI,CELL,XSCALE,YSCALE; REAL ACC; OWN REAL A,SC    11
B; OWN INTEGER LOW,HIGH; LABEL LAB1,LAB2,NXV,ERR,END1; SETSCALE(XSCALE,YSC    12
SCALE,1); IF TEST≠0 THEN GO TO NXV; COMMENT IF TEST≠0 THEN THE PURPOSE ISC    13
S TO SCALE A VALUE AND RETURN IT IN SCALED FORM IN RASTER COUNTS; LOW ← SC    14
LO ← ML; HIGH ← HI ← (1023-MR); IF XSCALE≠0 THEN BEGIN XL ← XMODV(XL); XSC    15
R ← XMODV(XR) END; A ← XR - XL; A ← (HIGH - LOW)/A; LAB1: B ← LOW - A×XLSC    16
; ACC ← SIGN(XL×A)×ENTIER(ABS(XL×A)+0.5); ACC ← SIGN(ACC+B)×ENTIER(ABS(ASC    17
CC+B)+0.5); ACC ← SIGN(ACC-LOW)×ENTIER(ABS(ACC-LOW)+0.5); LOW ← LOW - ACSC    18
C; ACC ← SIGN(XR×A)×ENTIER(ABS(XR×A)+0.5); ACC ← SIGN(ACC+B)×ENTIER(ABS(SC    19
ACC+B)+0.5); ACC ← SIGN(ACC-HIGH)×ENTIER(ABS(ACC-HIGH)+0.5); HIGH ← HIGHSC    20
 - ACC; A ← XR - XL; A ←(HIGH - LOW)/A; LAB2: B ← LOW - A×XL; LOW ← LO -SC    21
 1; HIGH ← HI + 1; GO TO END1; NXV: ERROR ← 0; IF XSCALE≠0 THEN IM ← XMOSC    22
DV(IM); ACC ← SIGN(IM×A)×ENTIER(ABS(IM×A)+0.5); ACC ← SIGN(ACC+B)×ENTIERSC    23
(ABS(ACC+B)+0.5); IF (ACC ≤ LOW)OR(ACC ≥ HIGH) THEN GO TO ERR; OUD ← ACCSC    24
; GO TO END1; ERR: ERROR ← 1; OUD ← 0; END1: END; PROCEDURE YSCALV(YB,YTSC    25
,MB,MT,TEST,IM,OUD,ERROR); VALUE YB,YT,MB,MT,TEST,IM; INTEGER MB,MT,TESTSC    26
,OUD,ERROR; REAL YB,YT,IM; BEGIN INTEGER LO,HI,CELL,XSCALE,YSCALE; REAL SC    27
ACC; OWN REAL A,B; OWN INTEGER LOW,HIGH; LABEL LAB1,LAB2,NYV,ERR,END1; SSC    28
ETSCALE (XSCALE, YSCALE,1); IF TEST ≠0 THEN GO TO NYV; LOW ← LO ← MB; HISC    29
GH ← HI ← (1023 -MT); IF YSCALE ≠ 0 THEN BEGIN YB ← YMODV(YB); YT←YMODV(SC    30
```

```
YT) END; A←YT-YB; A ← (HIGH - LOW)/A; LAB1: B ← LOW - A×YB; ACC ← SIGN(YSC      31
B×A)×ENTIER(ABS(YB×A)+0.5); ACC ← SIGN(ACC+B)×ENTIER(ABS(ACC+B)+0.5); ACSC      32
C ← SIGN(ACC-LOW)×ENTIER(ABS(ACC-LOW)+0.5); LOW ← LOW - ACC; ACC ← SIGN(SC      33
YT×A)×ENTIER(ABS(YT×A)+0.5); ACC ← SIGN(ACC+B)×ENTIER(ABS(ACC+B)+0.5); ASC      34
CC ← SIGN(ACC-HIGH)×ENTIER(ABS(ACC-HIGH)+0.5); HIGH ← HIGH - ACC; A ← YTSC      35
-YB; A ←(HIGH-LOW)/A; LAB2: B ← LOW - A×YB; LOW ← LO- 1;HIGH ← HI+ 1;GO SC      36
TO END1; NYV: ERROR ← 0; IF YSCALE≠0 THEN IM ← YMODV(IM); ACC ← SIGN(IM×SC      37
A)×ENTIER(ABS(IM×A)+0.5); ACC ← SIGN(ACC+B)×ENTIER(ABS(ACC+B)+0.5); IF (SC      38
ACC≤LOW)OR(ACC≥HIGH) THEN GO TO ERR; OUD ← ACC; GO TO END1; ERR: ERROR ←SC      39
 1; OUD ←0; END1: END; PROCEDURE DXDYV(IND,XL,XU,DX,N,I,NX,DC,KERR,MTB,MSC      40
TT); VALUE IND,XL,XU,MTB,MTT, DC; INTEGER IND,N,I,NX,KERR,MTB,MTT; REAL SC      41
XL,XU,DX,DC; BEGIN LABEL V2,V3,V4,V5,V6,V7,V8,V9,V10,V12,V13,V15,V16,V17SC      42
,V18, V19,V20,V21,V22,V23,V24,V25,V27,V30,V32,V33,V35,V36, V37,V38; SWITSC      43
CH SW1 ← V24,V19,V21,V21,V22,V23,V23,V23,V23,V24,V23; INTEGER MTL,MTR,KKSC      44
,NDIGT,LL,NLAB,ITIMS,NLINES,LABLE; REAL ALINES,ADX,POWR,DIGIT,XMAX,RT,COSC      45
UNTS,DELMRG,XDIF,XI; MTL ← 24; MTR ← 0; DELMRG ← 1023 - MTL - MTR -60; ISC      46
F (IND-1)<0 THEN GO TO V6 ELSE IF (IND-1)=0 THEN GO TO V3; V2: DELMRG ← SC      47
1023-MTB-MTT-60; V3: KERR←0; ITIMS ← 10; V4: XDIF ← ABS(XU-XL); IF XDIF≤SC      48
0 THEN GO TO V6; V5: NLINES ← DELMRG/DC; IF NLINES>0 THEN GO TO V7; V6: SC      49
KERR←1; GO TO V36; V7: ALINES ← NLINES; V8: ADX ← XDIF/ALINES; V9: IF (ASC      50
DX-0.999999)<0 THEN GO TO V13; V10: FOR KK ← 1 STEP 1 UNTIL 20 DO IF (ADSC      51
X-10)<0 THEN GO TO V16 ELSE ADX ← ADX/10; V12: GO TO V6; V13: FOR KK ←1 SC      52
STEP 1 UNTIL 20 DO IF (ADX-0.99999999)>0 THEN GO TO V15 ELSE ADX ← ADX×1SC      53
0; GO TO V6; V15: KK ← -(KK-2); V16: POWR ← 10↑(KK-1); NDIGT ← ADX + 0.9SC      54
; IF NDIGT<0 THEN GO TO V6 ELSE IF NDIGT=0 THEN GO TO V17 ELSE GO TO V18SC      55
; V17: NDIGT ← 1; V18: DIGIT ← NDIGT; DX ← DIGIT×POWR; N ← 5; I ← 10; GOSC      56
 TO SW1[NDIGT]; V19: DX ← POWR × 2; V20: I ← 5; GO TO V24; V21: DX ← POWSC      57
R × 5; V22: N ← I ← 4; GO TO V24; V23: DX ← POWR × 10; V24: IF ABS(XU)< SC      58
ABS(XL) THEN XMAX ← ABS(XL) ELSE XMAX ← ABS(XU); V25: XI ← I; RT ← DX × SC      59
XI; FOR LL ← 1 STEP 1 UNTIL 7 DO BEGIN NX ← LL; IF (RT-0.9999999)≥0 THENSC      60
 GO TO V27 ELSE RT ← RT×10 END; V27: IF (XMAX - 1)<0 THEN GO TO V32 ELSESC      61
 IF (NX-1)=0 THEN GO TO V30 ELSE NX ← NX+1; V30: FOR LL ← NX STEP 1 UNTISC      62
L 7 DO BEGIN NX ←LL; IF (XMAX-10)<0 THEN GO TO V32 ELSE XMAX ← XMAX/10 ESC      63

ND; V32: NLAB ← (NX+2)×10+5; COUNTS ← (DELMRG×DX)/(XDIF); IF (COUNTS-DC-SC      64
```

```
1)≤0 THEN GO TO V35; V33: LABLE ← COUNTS × XI; IF (NLAB-LABLE)≤0 THEN GOSC    65
 TO V36 ELSE I ← I+I; IF (I-16)<0 THEN GO TO V24; V35: ALINES ← 0.90×ALISC    66
NES; ITIMS ← ITIMS-1; IF (ITIMS)≤0 THEN GO TO V6 ELSE GO TO V8; V36: RT SC    67
← XDIF/(DX× I); IF (RT-2)<0 THEN GO TO V37 ELSE GO TO V38; V37: I ← N; NSC    68
X ← NX+2; V38: END; PROCEDURE PLOTP(CONT1,AREG); VALUE CONT1,AREG; INTEGSC    69
ER CONT1,AREG; COMMENT.....ASSUME PRESET OF INTEGER Z ← 1. USES 1 BUFFERSC    70
 OF 132 WORDS CONT1 = 0 = PLOTP CONT1 = 1 = PCLEAN; BEGIN INTEGER J,K,CTSC    71
1,CT2,CT3; LAREL PCLEAN,GOW,EXIT; DEFINE Z=TOMJIMJOEJOHN#,Q=THISISIDENTQSC    72
#; STREAM PROCEDURE EDITATAPE(Q,CT2,CT3); VALUE CT2,CT3; BEGIN SI := Q; SC    73
SI := SI + 10; DI := Q; CT2(4(DS := 6 CHR; SI := SI + 2)); CT3(4(DS := 6SC    74
 CHR; SI := SI + 2)) END; IF CONT1 = 1 THEN GO TO PCLEAN ELSE Q[Z] ← ARESC    75
G; IF Z ≠ 500 THEN GO TO EXIT ELSE GO TO GOW; PCLEAN: IF Z ≠ 500 THEN BESC    76
GIN J := Z MOD 4; FOR K := 1 STEP 1 UNTIL (4-J) DO BEGIN Q[Z] := 0; Q[Z]SC    77
.[12:6] := 10; Z := Z + 1 END; END; Q[Z] := AREG; GOW: CT1 := Z/4; CT3 :SC    78
= CT2 := 0; IF CT1 ≥ 63 THEN BEGIN CT2 := 63; CT3 := CT1 - 63 END ELSE CSC    79
T3 := CT1; EDITATAPE(Q,CT2,CT3); WRITE(FT4020,(Z×3)/4,Q[*]); Z := 0; EXISC    80
T: Z ← Z + 1 END; INTEGER PROCEDURE B5CODE(N); VALUE N; INTEGER N; BEGINSC    81
 B5CODE:= DRATTYA[N] END; PROCEDURE APLOTV(N,X,Y,IX,IY,NC,CHAR,IERR); VASC    82
LUE N, IX, IY, NC; INTEGER ARRAY CHAR[1]; REAL ARRAY X[1],Y[1]; INTEGER SC    83
N, IX, IY, NC, IERR; BEGIN INTEGER I,J,K,P,XERR,YERR,Q,R; LABEL APV1,APVSC    84
2,APV3; N ← ABS(N); IX ← ABS(IX); IY ← ABS(IY); K ← J ← 1; IERR ← 0; IF SC    85
N = 0 THEN GO TO APV1; COMMENT LOOP BEGINS HERE; FOR I := 1 STEP IX UNTISC    86
L N×IX DO BEGIN P←0; XSCALV(0,0,0,0,1,X[I],Q ,XERR);P.[20:10]:=Q; YSCALVSC    87
(0,0,0,0,1,Y[J],R ,YERR); P.[38:10] ← 1023 - R ; IF (XERR = 1)OR(YERR = SC    88
1) THEN GO TO APV2; P.[30:6] ← B5CODE(CHAR[K]); PLOTP(0,P); IF K=NC THENSC    89
 K ← 1 ELSE K ← K + 1; GO TO APV3; APV2: IERR ← IERR + 1; APV3: J ← J+IYSC    90
; END; APV1: END; PROCEDURE SIG6V(D,DBCD,NDX); VALUE D; REAL D; INTEGER SC    91
DBCD,NDX; COMMENT.....SIG6V WILL PICK 6 MOST SIGNIFICANT DIGITS AND STORSC    92
E IN DBED. SCALE FACTOR WILL BE STORED IN NDX; COMMENT.....SIG6V WILL NOSC    93
T DETERMINE SIGN; BEGIN INTEGER ARRAY NUT[1:6]; INTEGER SCALE,POWER,I; LSC    94
ABEL START,GOSIG,COMPUT; SCALE ← 6; POWER ← 5; DBCD ← 0; D ← ABS(D); D :SC    95
= D + 0.0000005; START: IF ENTIER(D) > 999999 THEN BEGIN D ← D/10; SCALESC    96
 ← SCALE + 1; GO TO START END ELSE GOSIG: IF (D/10↑POWER) ≥ 1 THEN GO TOSC    97
 COMPUT ELSE POWER ← POWER - 1; SCALE ← SCALE - 1; IF POWER ≠ -1 THEN GOSC    98
```

```
TO GOSIG ELSE COMPUT; FOR I ← 1 STEP 1 UNTIL 6 DO BEGIN NUT[I] ← ENTIERSC  99
(D/10*POWER) ; D := D -NUT[I] × 10*POWER; POWER := POWER - 1 END; NDX ← SC  100
SCALE; DBCD.[12:6] ← NUT[1].[42:6]; DBCD.[18:6] ← NUT[2].[42:6]; DBCD.[2SC  101
4:6] ← NUT[3].[42:6]; DBCD.[30:6] ← NUT[4].[42:6]; DBCD.[36:6] ← NUT[5].SC  102
[42:6]; DBCD.[42:6] ← NUT[6].[42:6] END; PROCEDURE PRINTV(N,A,NX,NY); VASC  103
LUE N,NX,NY; INTEGER N,NX,NY; ALPHA ARRAY A[1]; COMMENT TYPING STARTS ATSC  104
 CURRENT POINT IF BOTH NX AND NY ARE 1023, OTHERWISE TYPING BEGINS AT (NSC  105
X,NY); BEGIN INTEGER STOCOUNT, I, AREG,INDX,STORAG; LABEL CURPT, DUAL, SSC  106
PECPT, PLOT,START,FINIS; IF N=0 THEN GO TO FINIS; INDX:=0; AREG:=0; STOCSC  107
OUNT ← N MOD 6; IF STOCOUNT=0 THEN GO TO START; INDX:=I:=N DIV 6 + 1; STSC  108
ORAG:=A[I]; IF STOCOUNT=5 THEN A[I].[12:30] ← A[I].[18:30] ELSE IF STOCnSC  109
UNT=4 THEN A[I].[12:24] ← A[I].[24:24] ELSE IF STOCOUNT=3 THEN A[I].[12:SC  110
18] ← A[I].[30:18] ELSE IF STOCOUNT=2 THEN A[I].[12:12] ← A[I].[36:12] ESC  111
LSE IF STOCOUNT=1 THEN A[I].[12:6] ← A[I].[42:6]; START: IF NX = NY AND SC  112
NY = 1023 THEN GO TO CURPT ELSE GO TO SPECPT; CURPT: AREG.[12:30] := "B SC  113
   "; AREG.[42:6] ← B5CODE(A[1].[12:6])); PLOTP(0,AREG); I←1; STOCOUNT ←1SC  114
; IF N=1 THEN BEGIN AREG.[12:6] ← 10; GO TO PLOT END; DUAL: AREG.[12:6] SC  115
← B5CODE(A[I].[18:6])); STOCOUNT ← STOCOUNT + 1; IF STOCOUNT = N THEN BEGSC  116
IN AREG.[18:6] ← 10; GO TO PLOT END; AREG.[18:6] ← B5CODE(A[I].[24:6])); SC  117
STOCOUNT ← STOCOUNT + 1; IF STOCOUNT = N THEN BEGIN AREG.[24:6] ← 10; GOSC  118
 TO PLOT END; AREG.[24:6] ← B5CODE(A[I].[30:6])); STOCOUNT ← STOCOUNT + 1SC  119
; IF STOCOUNT = N THEN BEGIN AREG.[30:6] ← 10; GO TO PLOT END; AREG.[30:SC  120
6] ← B5CODE(A[I].[36:6])); STOCOUNT ← STOCOUNT + 1; IF STOCOUNT = N THEN SC  121
BEGIN AREG.[36:6] ← 10; GO TO PLOT END; AREG.[36:6] ← B5CODE(A[I].[42:6]SC  122
)); STOCOUNT ← STOCOUNT + 1; IF STOCOUNT = N THEN BEGIN AREG.[42:6] ← 10;SC  123
 GO TO PLOT END; I ← I+1; AREG.[42:6] ← B5CODE(A[I].[12:6])); PLOTP(0,ARESC  124
G)); STOCOUNT ← STOCOUNT + 1; IF STOCOUNT = N THEN BEGIN AREG.[12:6] ← 10SC  125
; GO TO PLOT END; GO TO DUAL; SPECPT: AREG ← 0; AREG.[12:6] ← 16; AREG.[SC  126
20:10] ← NX; AREG.[38:10] ← 1023 - NY; AREG.[30:6] ← B5CODE(A[1].[12:6]))SC  127
; PLOTP(0,AREG)); I ← 1; STOCOUNT ← 1; IF STOCOUNT = N THEN BEGIN AREG.[1SC  128
2:6] ← 10; GO TO PLOT END; GO TO DUAL; PLOT: PLOTP(0,AREG));IF INDX=0 THESC  129
N GO TO FINIS; A[INDX]:=STORAG; FINIS:END; PROCEDURE ERMRKV; BEGIN INTEGSC  130
ER I1,W1,W2,AREG; COMMENT,,,,,PLACE ERROR MARKS ON FRAME; W1 ← 24014397SC  131
6; W2 ← 1048576; AREG ← W1; FOR I1 ← 1 STEP 1 UNTIL 10 DO BEGIN PLOTP(0,SC  132
```

```
AREG)} AREG ← AREG + W? END END} PROCEDURE RESETV} COMMENT.....OCT 56, ASC  133
DVANCE FILM, STOP TYPE, AND SET HIGH INTENSITY} BEGIN INTEGER AREG} AREGSC  134
 ← 0} AREG.[12:6]←46} PLOTP(1,AREG) END} PROCEDURE INTENS(MODE)} VALUE MSC  135
ODE} ALPHA MODE} COMMENT.....MODE = "HIGH" OR "LOW" WILL PLOT BLANK AT OSC  136
,0} BEGIN INTEGER AREG} AREG := 0} AREG.[12:6] := 2} AREG.[30:6] := 48} SC  137
IF MODE = "LOW" THEN AREG.[12:6] := 4} PLOTP(0,AREG) END} PROCEDURE FORMSC  138
V} COMMENT.....OCTAL 50 COMMAND, PROBABLY NOT USED} BEGIN INTEGER AREG} SC  139
AREG := 0} AREG.[12:6] ← 40} PLOTP(1,AREG) END} PROCEDURE ERRNLV(XYL,XYUSC  140
,ML,MU,DXY)} REAL DXY,XYL,XYU} INTEGER ML,MU} COMMENT.....DOES SAME FOR SC  141
NONLINEAR PLOT AS ERRLNV DOES FOR LINEAR} BEGIN INTEGER ERRORMARK} REAL SC  142
CELL} LABEL ER10,ER20,ER21,ER30,ER40, ER60,ER80,EREND,ER50} IF(MU + ML -SC  143
 1023) < 0 THEN GO TO ER10 ELSE ERRORMARK ← 1} MU ← ML ← 500} ER10: IF XSC  144
YL ≠ 0 THEN BEGIN CELL ← XYL} GO TO ER30 END ELSE IF XYU = 0 THEN BEGIN SC  145
CELL ← 1.0} GO TO ER21 END ELSE CELL ← 0.1 × XYU} ER20: CELL ← ABS(CELL)SC  146
} ER21: ERRORMARK ← 1} ER30: IF CELL <0 THEN GO TO ER20 ELSE XYL ← CELL}SC  147

 IF XYU ≠ 0 THEN BEGIN CELL ← XYU} GO TO ER50 END ELSE CELL ← 10.0 × XYLSC  148
} ER40: CELL ← ABS(CELL)} ERRORMARK ← 1} ER50: IF CELL < 0 THEN GO TO ERSC  149
40 ELSE XYU ← CELL} IF(XYU - XYL) ≠ 0 THEN GO TO ER60 ELSE ERRORMARK ← 1SC  150
} XYL ← 0.1 × XYU} ER60: CELL ← (10 ↑ 10) × XYL} IF CELL ≥ XYU THEN GO TSC  151
O ER80 ELSE XYU ← CELL} ERRORMARK ← 1.0} ER80: IF ERRORMARK ≠ 0 THEN ERMSC  152
RKV} EREND: END} PROCEDURE ERRLNV(XYL,XYU,ML,MU,DXY)} COMMENT THIS PROCESC  153
DURE CHECKS VALIDITY FOR THE SPECIFIED PARAMETER VALUES. IF THE PARAMETESC  154
RS ARE INVALID, A PRESET VALUE WILL REPLACE THE ERRONEOUS VALUE} REAL XYSC  155
L,XYU,DXY} INTEGER ML, MU} BEGIN REAL CELL,CELL2} INTEGER ERRORMARK} LABSC  156
EL EC,EA,EB,EE,EF,EG} ERRORMARK ← 0} CELL ← XYU - XYL} IF CELL=0 THEN GOSC  157
 TO EC ELSE GO TO EA} EC: XYU ← XYL ← 10.0} ERRORMARK ← 1} CELL ← 10} EASC  158
: IF (ML+MU - 1023) <0 THEN BEGIN CELL2 ← ABS(ML+MU-1023)} GO TO EB END}SC  159

 ML ← MU ← 500} CELL2 ← 23} ERRORMARK ← 1} EB: CELL ← ABS(CELL2)/CELL} CSC  160
OMMENT IF DIVIDE CHECK OCCURS GO TO EC} IF (ABS(CELL × DXY) -3.0)>0 THENSC  161
 GO TO EE} DXY ← 0} ERRORMARK ← 1} EE: IF ERRORMARK = 1 THEN GO TO EF ELSC  162
SE GO TO EG} EF: ERMRKV} EG: END} PROCEDURE LABLV(D,IX,IY,NX,INT,NDMAX)}SC  163
 VALUE D,IX,IY,NX,INT,NDMAX} INTEGER IX,IY,NX,INT,NDMAX} COMMENT..... D SC  164
= FLOATIMG POINT NO. TO CONVERT, IX,IY = ORDINATES (IN RASTER COUNTS) FOSC  165
```

```
R LOC OF 1ST CHAR. THESE ORDINATES WILL BE USED FOR CENTER OF CHAR. NX =SC   166
 NO. CHARACTERS TO BE DISPLAYED NT = NO. TIMES TO DISPLAY THESE CHAR. INSC   167
 SAME POSITION NDMAX = MAXIMUM DECIMAL SCALE; REAL D; COMMENT....USES PRSC   168
INTV,CHARACTRON CHARACTERS; BEGIN ALPHA BLNKO,BLANK,MIN,DECP,DECPT,B1,B2SC   169
,B3,B4,B5 ; ALPHA ARRAY DBCD[1:2]; LABEL LB20,LB30,LB40,LB41,LB60,I5,COMSC   170
1, LB21,LB61,LB62; INTEGER I1,MAXNX,J,NDX,AREG,MQ,QREG,ACC,WORK1,ZERO,I2SC   171
,CT; BLNKO←"      0"; BLANK←"       "; MIN←"-00000"; DECP←".00000"; DECPT←SC   172
"00000."; B1←"0      "; B2←"00     "B3←"000    "; B4←"0000   "; B5←"00000 "SC   173
; J ← 1; MQ ← QREG ← ACC ← WORK1 ← ZERO ← CT ← 0; I2 ← 6; MIN ← "-00000"SC   174
; MAXNX ← 6; SIG6V(D,DBCD[J],NDX)] QREG ← DBCD[1]; I1 ← NDMAX; IF NDMAX SC   175
> NDX THEN BEGIN I1 ← I1 - NDX; AREG ← BLNKO END ELSE GO TO LB21; IF NDXSC   176
 ≠ 0 THEN AREG ← BLANK; IF I1 > 6 THEN GO TO LB41 ELSE IF I1 = 1 THEN BESC   177
GIN MQ.[18:30] ← QREG.[12:30]; MQ.[12:6] ← AREG.[42:6]; ACC ← B1; END ELSC   178
SE IF I1 = 2 THEN BEGIN MQ.[24:24] ← QREG.[12:24]; MQ.[12:12] ← AREG.[36SC   179
:12]; ACC ← B2; END ELSE IF I1 = 3 THEN BEGIN MQ.[30:18] ← QREG.[12:18];SC   180
 MQ.[12:18] ← AREG.[30:18]; ACC ← B3 END ELSE IF I1 = 4 THEN BEGIN MQ.[3SC   181
6:12] ← QREG.[12:12]; MQ.[12:24] ← AREG.[24:24]; ACC ← B4 END ELSE IF I1SC   182
 = 5 THEN BEGIN MQ.[42:6] ← QREG.[12:6]; MQ.[12:30] ← AREG.[18:30]; ACC SC   183
← B5 END; LB20:AREG ← ACC; QREG ← MQ; LB21: I1 := NDMAX; IF NDMAX ≤ 5 THSC   184
EN GO TO LB30 ELSE WORK1 ← AREG; AREG ← QREG; QREG ← WORK1; IF NDMAX > 6SC   185
 THEN GO TO LB41 ELSE QREG ← DECP; GO TO LB40; LB30:AREG ← 0; MQ ← 0; IFSC   186
 I1 = 0 THEN BEGIN AREG.[12:6] ← DECPT.[42:6]; AREG.[18:30] ← QREG.[12:3SC   187
0] END ELSE IF I1 = 1 THEN BEGIN AREG.[12:6] ← QREG.[12:6]; AREG.[18:6] SC   188
← DECPT.[42:6]; AREG.[24:24] ← QREG.[18:24] END ELSE IF I1 = 2 THEN BEGISC   189
N AREG.[12:12] ← QREG.[12:12]; AREG.[24:6] ← DECPT.[42:6]; AREG.[30:18] SC   190
← QREG.[24:18] END ELSE IF I1 = 3 THEN BEGIN AREG.[12:18] ←QREG.[12:18];SC   191
 AREG.[30:6] ← DECPT.[42:6]; AREG.[36:12] ← QREG.[30:12] END ELSE IF I1 SC   192
= 4 THEN BEGIN AREG.[12:24] ← QREG.[12:24]; AREG.[36:6] ← DECPT.[42:6]; SC   193
AREG.[42:6] ← QREG.[36:6] END ELSE I5: BEGIN AREG.[12:30] ← QREG.[12:30]SC   194
; AREG ← AREG ← DECPT END; COM1:MQ.[12:6] ← QREG.[42:6]; QREG ← MQ; LB40SC   195
:I2 ← I2 + 1; MAXNX ← MAXNX + 1; IF D ≥ 0 THEN GO TO LB60; IX ← IX - 8; SC   196
I1 ← I1 + 1; LB41:DBCD[1] ← AREG; I1 ← NX; IF I1 > MAXNX THEN I1 ← MAXNXSC   197
; QREG.[18:30]← QREG.[12:30]; QREG.[12:6] ← AREG.[42:6]; AREG.[18:30] ← SC   198
AREG.[12:30]; AREG.[12:6] ← ZERO.[12:6]; DBCD[1] ← AREG ← MIN; LB60:DBCDSC   199
```

```
[2] ● QREG; NX ● I1; DBCD[2].[42:6] := DBCD[2].[12:6]; IF (NX = 0) OR (NSC   200
X = 6) THEN GO TO LB61 ELSE IF NX = 1 THEN DBCD[1].[42:6] := DBCD[1].[12SC   201
:6] ELSE IF NX = 2 THEN DBCD[1].[36:12] := DBCD[1].[12:12] ELSE IF NX = SC   202
3 THEN DBCD[1].[30:18] := DBCD[1].[12:18] ELSE IF NX = 4 THEN DBCD[1].[2SC   203
4:24] := DBCD[1].[12:24] ELSE IF NX = 5 THEN DBCD[1].[18:30] := DBCD[1].SC   204
[12:30]; LB61: FOR I1:= 1 STEP 1 UNTIL INT DO PRINTV(NX,DBCD ,IX,IY); LBSC   205
62: END; PROCEDURE SCRND(XMAX, XMIN, XRMX,XRMN); VALUE XMAX,XMIN; COMMENSC   206
T GIVEN EXTREME VALUES OF A VARIABLE TO BE PLOTTED, PRACTICAL GRID LIMITSC   207
S ARE DETERMINED. VARIABLE MAY BE X OR Y; REAL XMAX, XMIN,XRMX,XRMN; BEGSC   208
IN REAL XR1, XL1,DX,SC1,SC2, XR, XL, Z; INTEGER NCNT, KS; LABEL SC30, SCSC   209
40, SC60, SC80, SC100, SC110, SC140, SC150, SC180, SC190, SC200, SC210, SC   210
SC220,SC170; XR1 ← XMAX; XL1 ● XMIN; DX ← XR1 - XL1; IF DX ≤ 0 THEN GO TSC   211
O SC170; NCNT ← 0; SC1 ● KS ← 1; IF ABS(XR1) - ABS(XL1) ≥0 THEN GO TO SCSC   212
30; KS ← -1; XR1 ← -XL1; XL1 ← -XMAX; COMMENT NOW XR1 MUST BE POSITIVE, SC   213
AND GREATER IN MAGNITUDE THAN XL1; SC30: SC2 ← SC1×10; XR ← XR1; XL ← XLSC   214
1; Z←1; SC40: IF (XR -SC1)=0 THEN GO TO SC110 ELSE IF (XR - SC1)>0 THEN SC   215
GO TO SC60; XR ← XR × 10; Z ← Z × 10; GO TO SC40; SC60: IF (XR - SC2)<0 SC   216
THEN GO TO SC80 ELSE IF (XR - SC2) = 0 THEN GO TO SC100; XR ← XR/10; Z ←SC   217
 Z/10; GO TO SC60; COMMENT XR NOW LIES BETWEEN SC1 AND SC1 × 10; SC80: XSC   218

R ← SIGN(XR)× ENTIER(ABS(XR)); IF (XR - XR1 × Z) ≥0 THEN GO TO SC100; XRSC   219
 ● XR+1; SC100: XR ← XR/Z; COMMENT ROUND XL, WHICH MAY BE POS OR NEG. NOSC   220
TE THAT ABS(XL × Z) MUST BE LESS THAN SC1 × 10, WHICH IS BOUNDED BY (FORSC   221
 NCNT =4) 100,000; SC110: XL ← SIGN(XL × Z)×ENTIER(ABS(XL × Z)); IF (XL1SC   222
)=0 THEN GO TO SC150 ELSE IF (XL1)>0 THEN GO TO SC140; IF (XL-XL1 × Z) ≤SC   223
0 THEN GO TO SC140; XL ← XL -1; SC140: XL ← XL/Z; SC150: IF (DX/(XR-XL)-SC   224
.8) ≥0 THEN GO TO SC190; NCNT ← NCNT+1; IF (NCNT-4)≤0 THEN GO TO SC180; SC   225
SC170: XRMX ← XMAX; XRMN ← XMIN; GO TO SC220; SC180: SC1 ← SC1 × 10; GO SC   226
TO SC30; SC190: IF (KS)>0 THEN GO TO SC210; SC200: XRMX ← -XL; XRMN ← -XSC   227
R; GO TO SC220; SC210: XRMX ← XR; XRMN ← XL; SC220: END; PROCEDURE LINEVSC   228
 (NX0,NY0, NX1, NY1); VALUE NX0,NY0,NX1,NY1; INTEGER NX0, NY0, NX1, NY1;SC   229
 BEGIN DEFINE Z = TOMJIMJOEJOHN#, Q = THISISIDENTQ#; INTEGER AREG, XDIFFSC   230
,YDIFF, XCURRENT, YCURRENT,CELL, XREM, YREM; LABEL LN1, LN2, LN3, LNEND;SC   231
 NX0←ABS(NX0)MOD(1024);NX1←ABS(NX1)MOD(1024); NY0 ← 1023 - NY0; NY1 ● 10SC   232
23 - NY1; XDIFF ← (NX1-NX0); YDIFF ← (NY0-NY1); IF (XDIFF=0) AND (YDIFF S0   233
```

```
=0) THEN GO TO LNEND; XCURRENT ← NX0; YCURRENT ← NY0; XREM ← ABS(XDIFF);SC   234
 YREM ← ABS(YDIFF); LN1: AREG ← 0; AREG.[12:2] ← 3; IF XDIFF ≥0 THEN ARESC   235
G.[30:1] ← 1; IF YDIFF ≥0 THEN AREG.[31:1] ← 1; AREG.[20:10] ← XCURRENT.SC   236
[38:10]; AREG.[38:10] ← YCURRENT.[38:10]; IF (XREM≤63) AND (YREM≤63) THESC   237
N BEGIN AREG.[14:6] ← XREM.[42:6]; AREG.[32:6] := YREM.[42:6]; IF Z = 50SC   238
0 THEN PLOTP(0,AREG) ELSE BEGIN Q[Z] := AREG; Z := Z + 1 END; GO TO LNENSC   239
D END ELSE IF XREM ≥ YREM THEN GO TO LN2 ELSE GO TO LN3; LN2: AREG.[14:6SC   240
] ← 63; CELL ← ENTIER((63×YREM)/XREM); AREG.[32:6] := CELL.[42:6]; IF Z SC   241
= 500 THEN PLOTP(0,AREG) ELSE BEGIN Q[Z] := AREG; Z := Z + 1 END; XCURRESC   242
NT ← XCURRENT + SIGN(XDIFF)×63; YCURRENT ← YCURRENT - SIGN(YDIFF)×CELL; SC   243
YREM ← YREM - CELL; XREM ← XREM - 63; GO TO LN1; LN3: AREG.[32:6] ← 63; SC   244
CELL ← ENTIER((63×XREM)/YREM); AREG.[14:6] := CELL.[42:6]; IF Z = 500 THSC   245
EN PLOTP(0,AREG) ELSE BEGIN Q[Z] := AREG; Z := Z + 1 END; XCURRENT ← XCUSC   246
RRENT + SIGN(XDIFF)×CELL; YCURRENT ← YCURRENT - SIGN(YDIFF)×63; XREM ← XSC   247
REM - CELL; YREM ← YREM -63; GO TO LN1; LNEND: END; PROCEDURE CAMRAV(N);SC   248
 VALUE N; INTEGER N; BEGIN INTEGER AREG,TEMP; LABEL CX; AREG ← 0; IF N =SC   249
 35 THEN BEGIN TEMP ← 33; GO TO CX END ELSE IF N = 9 THEN BEGIN TEMP ← 3SC   250
4; GO TO CX END ELSE TEMP ← 35; CX: AREG.[12:6] ← TEMP; PLOTP(1,AREG) ENSC   251
D; PROCEDURE PLOTV(NX,NY,K,J); VALUE J,NX,NY,K; COMMENT NX AND NY MAY BESC   252
 OF TYPE REAL OR INTEGER DEPENDING ON WHETHER J IS / OR 0. IF J=0 THEN NSC   253
X AND NY ARE RASTER COUNTS; INTEGER K, J; REAL NX,NY; BEGIN INTEGER STORSC   254
E, X, Y, CHEK1; LABEL OTHER; IF J≠0 THEN BEGIN XSCALV(0,0,0,0,1,NX,X, CHSC   255
EK1); YSCALV(0,0,0,0,1,NY,Y,CHEK1); Y ← 1023 - Y; GO TO OTHER END; X ←(NSC   256
X)MOD(1024); Y ← 1023 -(NY)MOD(1024); OTHER: STORE ← 0; STORE.[20:10] ← SC   257
X; STORE.[38:10] ← Y; STORE.[30:6] ← B5CODE(K.[42:6]); PLOTP(0, STORE) ESC   258
ND; PROCEDURE NONLNV(I,LREFR,IBASE,ITOP,XYL,XYU,DXY,NM,IJ,NXY,IWH); VALUSC   259
E I,LREFR,IBASE,ITOP,XYL,XYU,DXY,NM,IJ,NXY,IWH; INTEGER I,LREFR,IBASE,ITSC   260
OP,NM,IJ,NXY,IWH; REAL XYL, XYU, DXY; BEGIN LABEL L409,L410,L414,L417,L4SC   261
21,L423,L424,L430,L431,L436,L437, L438,L439,L441,L442,L445,L446,L448; RESC   262
AL DELTA, A,B,XYMAX,XYMIN,START,RS,XY,STOP; INTEGER IY1,IY2,IYL,IX1,IX2,SC   263
IXL,NDMAX,L,II; IX1 ← IY1 ← IBASE; IX2 ← IY2 ← ITOP; IXL ← IYL ← LREFR; SC   264
DELTA ← ABS(DXY); IF (DELTA-1.0)<0 THEN GO TO L448 ELSE IF DELTA=1.0 THESC   265
N GO TO L410; L409: DELTA ← 10; L410: A ← ABS(XYL); B ← ABS(XYU); IF A<BSC   266
```

```
 THEN BEGIN XYMAX ← B; XYMIN ← A END ELSE BEGIN XYMAX ← A; XYMIN ← B ENDSC   267
; L414: SIG6V(XYMAX,B,NDMAX); L:=LN(XYMIN)/LN(10); START:=10*L/10; L417:SC   268
 RS ← 1; II ← 1; XY ← START; STOP ← START×10; L421: IF I=2 THEN GO TO L4SC   269
30; L423: XSCALV(0,0,0,0,1,XY,IX1,0); IF (IX1)≤0 THEN GO TO L441; L424: SC   270
IX2 ← IX1; IXL ← IX1 -((NXY-1)×IWH)/2; GO TO L436; L430: YSCALV(0,0,0,0,SC   271
1,XY,IY1,0); IF (IY1)≤0 THEN GO TO L441; L431: IY2 ← IY1; IYL ← IY1 + 3 SC   272
+ IWH/2; L436: LINEV(IX1,IY1,IX2,IY2); IF II=2 THEN GO TO L442; L437: IFSC   273
 (NXY)=0 THEN GO TO L439; L438: LABLV(XY,IXL,IYL,NXY, 1,NDMAX); L439: IISC   274
 ← 2; GO TO L436; L441: II ← 2; L442: RS ← RS+DELTA; XY ← START×RS; IF (SC   275
XY-XYMAX)>0 THEN GO TO L448; L445: IF (XY-STOP)<0 THEN GO TO L421; L446:SC   276
 START ← STOP; GO TO L417; L448: END; PROCEDURE FRAMEV; COMMENT.....OCT S6   277
46 WILL ADVANCE FILM FRAME ONLY; BEGIN INTEGER AREG; AREG ← 0; AREG.[12:SC   278
6] ← 38; PLOTP(1,AREG) END; PROCEDURE LINRV (I,LREFR,IBASE,ITOP,XYL,XYU,SC   279
DXY,NM,IJ,NXY,IWH); VALUE I,LREFR,IBASE,ITOP,XYL,XYU,DXY,NM,IJ,NXY,IWH; SC   280
INTEGER I,LREFR,IBASE,ITOP,NM,IJ,NXY,IWH; REAL XYL,XYU,DXY; BEGIN LABEL SC   281
V3030,V3050,V3130,V3170,V3220,V3260,V3290,V3295,V3310, V3312,V3315,V3320SC   282
,V3330,V3332,V3350,V3400,V3410,V3460, V3494,V3496,V3498,V3500,V3510,V353SC   283
0,V3540,V3550,V3560, V3570,V3600,V3620,V3630,V3660,V3670,V3730,V3740; INSC   284
TEGER NABS, IABS, IX1, IX2,IX,IY,IY1,IY2,NDMAX, K, J, NCHAR; REAL DELTA,SC   285
 A,B,STOP1,STOP2,T,XY,START,TRIAL,XCOUNT,XYN; SWITCH SWT ← V3332,V3290,VS6   286
3310; V3030: DELTA ← ABS(DXY); IF DELTA=0 THEN GO TO V3740; V3050: NABS SC   287
← ABS(NM); IABS ← ABS(IJ); IX1 ← IBASE; IX2 ← ITOP; IX ← LREFR; IY1 ← IXSC   288
1; IY2 ← IX2; IY ← IX; A ← XYU; B ← XYL; IF A<B THEN BEGIN STOP1 ← B; STSC   289
OP2 ← A END ELSE BEGIN STOP1 ← A; STOP2 ← B END; IF ABS(A) < ABS(B) THENS6   290
 T←ABS(B) ELSE T ← ABS(A); SIG6V(T, XY, NDMAX); T ← A×B; IF T ≤0 THEN GOSC   291
 TO V3130 ELSE GO TO V3170; V3130: TRIAL ← START ← 0; GO TO V3260; V3170SC   292
: TRIAL ← STOP2; IF (STOP1) < 0 THEN TRIAL ← STOP1 ELSE IF STOP1 = 0 THESC   293
N GO TO V3130; V3220: T ← TRIAL/DELTA; IF (ABS(T)-9999999.0)≥0 THEN GO TSC   294
O V3130 ELSE START ← DELTA×SIGN(T)×ENTIER(ABS(T)); V3260: XCOUNT ← 1.0; SC   295
COMMENT BEGIN OUTER LOOP; FOR K ← 1 STEP 1 UNTIL 3 DO BEGIN GO TO SWT[K]SC   296
; V3290: IF NABS<0 THEN GO TO V3050 ELSE IF NABS=0 THEN GO TO V3730; V32SC   297
95: XCOUNT ← NABS; GO TO V3320; V3310: IF NXY=0 THEN GO TO V3730; V3312:SC   298
 IF IABS<0 THEN GO TO V3050 ELSE IF IABS=0 THEN GO TO V3730; V3315: XCOUSC   299
NT ← IABS; V3320: T ← DELTA × XCOUNT; V3330: START ← T×SIGN(TRIAL/T)×ENTSC   300
```

```
IER(ABS(TRIAL/T))} COMMENT BEGIN INNER LOOP} V3332: FOR J←1 STEP 1 UNTILSC   301
 2 DO BEGIN XYN ← 0} V3350: XY ← START + DELTA × XYN} V3400: IF I=2 THENSC   302
 GO TO V3460} V3410: XSCALV(0,0,0,0,1,XY,IX1,0)} IX2 ← IX1} IF IX1=0 THESC   303
N GO TO V3494 ELSE GO TO V3500} V3460: YSCALV(0,0,0,0,1,XY,IY1,0)} IY2 ←SC   304
 IY1} IF IY1=0 THEN GO TO V3494 ELSE GO TO V3500} V3494: IF J=2 THEN GO SC   305
TO V3498} V3496: IF (XY - STOP1)<0 THEN GO TO V3630 ELSE GO TO V3660} V3SC   306
498: IF (STOP2 - XY)<0 THEN GO TO V3630 ELSE GO TO V3660} V3500: IF K = SC   307
3 THEN GO TO V3530} V3510: LINEV(IX1,IY1,IX2,IY2)} GO TO V3630} V3530: NSC   308
CHAR ← NXY} V3540: IF XY=0 THEN GO TO V3550 ELSE GO TO V3560} V3550: IF SC   309
2 < NDMAX THEN NCHAR ← NDMAX ELSE NCHAR ← 2} V3560: IF I=2 THEN GO TO V3SC   310
600} V3570: IX ← IX1 - ((NXY-1)×IWH)/2} GO TO V3620} V3600: IY ← IY1 + (SC   311
IWH)/2 + 3} V3620: LABLV(XY,IX,IY,NCHAR, 1,NDMAX)} V3630: XYN ← XYN + XCSC   312

OUNT} GO TO V3350} V3660: DELTA ← -DELTA} V3670: END OF INNER LOOP} V373SC   313
0: END OF OUTER LOOP} V3740: END OF LINRV} PROCEDURE STOPTV} BEGIN INTEGSC   314
ER AREG} AREG := 0} AREG.[12:6] ← 10} PLOTP(0,AREG) END} PROCEDURE YAXISSC   315
V(NX,NY)} VALUE NX,NY} INTEGER NX,NY} BEGIN INTEGER AREG} AREG:=0} AREG.SC   316
[12:6] ← 26} AREG.[20:10] ← NX} AREG.[38:10] ← 1023 - NY} PLOTP(1,AREG) SC   317
END} PROCEDURE XAXISV(NX,NY)} VALUE NX,NY} INTEGER NX,NY} BEGIN INTEGER SC   318
AREG} AREG:=0} AREG.[12:6] ← 24} AREG.[20:10] ← NX} AREG.[38:10] ← 1023 SC   319
 - NY} PLOTP(1,AREG) END} PROCEDURE HOLLV(IJ,WORDS,L)} VALUE IJ} INTEGER SC   320
IJ,L} ALPHA ARRAY WORDS[1]} COMMENT....WORDS = BLOCK OF BCD CHAR. L = LOSC   321
C OF OUTPUT CHAR 0000000A COMMENT....IJ = NO. OF CHARACTER TO BE PICKED SC   322
COMMENT....ASSUME DATA IN ASCENDING SEQUENCE} BEGIN INTEGER INT,UNR,AREGSC   323
} LABEL EX1} INT ← IJ DIV 6} UNR ← IJ MOD 6} IF UNR ≠ 0 THEN INT ← INT +SC   324
1} AREG ← WORDS[INT].[12:36]} IF UNR = 0 THEN BEGIN L ← AREG.[42:6]} GO SC   325
TO EX1 END ELSE IF UNR = 1 THEN BEGIN L ← AREG.[12:6]} GO TO EX1 END ELSSC   326
E IF UNR = 2 THEN BEGIN L ← AREG.[18:6]} GO TO EX1 END ELSE IF UNR = 3 TSC   327
HEN BEGIN L ← AREG.[24:6]} GO TO EX1 END ELSE IF UNR = 4 THEN BEGIN L ← SC   328
AREG.[30:6]} GO TO EX1 END ELSE L ← AREG.[36:6]} EX1: END} PROCEDURE APRSC   329
NTV(INCRX,INCRY,NCHAR,WORDS,IX,IY)} VALUE IX,IY,NCHAR,INCRX,INCRY} INTEGSC   330
ER INCRX,INCRY,NCHAR,IX,IY} ALPHA ARRAY WORDS[1]} BEGIN INTEGER J,L} NCHSC   331
AR ← ABS(NCHAR)} FOR J ← 1 STEP 1 UNTIL NCHAR DO BEGIN HOLLV(J,WORDS,L)}SC   332
 PLOTV(IX,IY,L,0)} IX ← IX + INCRX} IY ← IY + INCRY END END} PROCEDURE MSC   333
ARGIN(L1,ML3,A,B)} VALUE L1} INTEGER L1,ML3,A,B} BEGIN INTEGER ARRAY IQ[SC   334
```

```
1:3], L[1:6,1:3]; LABEL L8,L100; INTEGER IC; OWN INTEGER K; IF L1 ≥ 0 THSC   335
EN GO TO L8 ELSE K ← ABS(L1); GRAPHCTPERPAGE := 1; L8: L[1,1] ← L[3,2] ←SC   336
 L[6,1] ← L[6,2] ← L[5,2] ← L[4,1] ← 24; L[1,2] ← L[3,1] ← 690; L[1,3] ←SC   337
 178; L[2,1] ← L[2,2] ← 357; L[2,3] ← 511; L[3,3]← 844; L[4,2] ← L[5,1] SC   338
← 524; L[4,3] ← 262; L[5,3] ← 762; L[6,3] ← 512; IQ[1] ← 5; IQ[2] ← 3; ISC   339
Q[3] ← 0; L100: IC := GRAPHCTPERPAGE + IQ[K]; ML3 ← L[IC,3]; A ← L[IC,1]SC   340
; B ← L[IC,2]; GRAPHCTPERPAGE := GRAPHCTPERPAGE + 1 END; INTEGER PROCEDUSC   341
RE NBLANK(WORD,N); VALUE N; INTEGER N; ALPHA ARRAY WORD[1]; COMMENT.....SC   342
NBLANK DETERMINES NO. OF CHARACTERS IN HOLLERITH LABEL NBLANK.....IS 0 ISC   343
F WORD IS BLANK NBLANK.....IS NO. OF CHARACTERS IF (WORD) IS NOT BLANK; SC   344
BEGIN INTEGER N1, I; LABEL LB1; N1 := 0; FOR I := N STEP -1 UNTIL 1 DO BSC   345
EGIN IF WORD[I] = "      " THEN N1 := N1 + 1 ELSE GO TO LB1 END; LB1: NBSC   346
LANK := (N-N1) × 6 END; PROCEDURE GRID1V(L,XL,XU,YL,YU,DX,DY,NN,MM,II,JJSC   347
,NX,NY,A,B); VALUE L,XL,XU,YL,YU,DX,DY,NN,MM,II,JJ,NX,NY,A,B; COMMENT...SC   348
..GEN. GRID SC-4020, LIN OR NONLIN ON EITHER × OR Y AXIS; INTEGER L,NN,MSC   349
M,II,JJ,NX,NY,A,B; REAL XU,XL,YL,YU,DX,DY; BEGIN REAL ARRAY XYL,XYU,DXY,SC   350
PRDUCT[1:2]; ALPHA ARRAY C[1:1]; INTEGER ARRAY NXY,K,ISPACE,LU,ML,MU,ITTSC   351
,MTL,MTU[1:2], NM,IJ[1:2], LL[1:22],LREFR[1:12]; DEFINE JOBID=THISISJOBISC   352
D#; DEFINE PAGECT=XYZWPAGECT#; REAL YMIN, AML, AMU; INTEGER IWD2,IHD2,ITSC   353
,MUH,IWIDE,IHIGH,I,MUV; LABEL L105,L110,LISPACE,L260,L350,L360,L370,L420SC   354
,L460,L480, L660,L668,L670,L690,L752,L756,L780,L790,L810,L850,L870, L910SC   355
,START,L120,L700,L930,L758; SWITCH SW1 ← L370,L420; SWITCH SW2 ← L752,L7SC   356
56;; COMMENT.....STOP TYPE MAY NOT BE NEEDED; START: STOPTV; INTENSC "HISC   357
GH")); IF (L-2) = 0 THEN GO TO L110; L105: FRAMEV; C[1] := "PAGE"; PRINTVSC   358
(4,C,940,1023); PAGECT := PAGECT + 1; LABLV(PAGECT,980,1023,3,1,3); MUV SC   359
:= 780; FOR I := 1 STEP 1 UNTIL 16 DO BEGIN C[1] := JOBID[I]; PRINTV(1,CSC   360
,MUV,1023); MUV := MUV + 8 END; L110: MTL[1] ← 24; MTU[1] ← 0; MTL[2] ← SC   361
A; MTU[2] ← B; IWIDE ← 8; IHIGH ← 10; SETSCALE(K[1],K[2],1); L120: XYL[1SC   362
] ← XL; XYL[2] ← YL; XYU[1] ← XU; XYU[2] ← YU; DXY[1] ← DX; DXY[2] ← DY;SC   363
 NM[1] ← NN; NM[2] ← MM; IJ[1] ← II; IJ[2] ← JJ; NXY[1] ← NX; NXY[2] ← NSC   364
Y; IWD2 ← IWIDE/2; IHD2 ← IHIGH/2; IF XYL[2] < XYU[2] THEN BEGIN YMIN ← SC   365
XYL[2]; GO TO LISPACE END ELSE YMIN ← XYU[2]; LISPACE: ISPACE[1] ← NXY[2S6   366
] × IWIDE + 6; IF YMIN < 0 THEN ISPACE[1] ← ISPACE[1] + IWIDE ; ISPACE[2SC   367
] ← IHIGH +6; L260: LU[1] ← NXY[1] × IWD2; COMMENT.....ZERO TEST REMOVEDSC   368
```

```
; LU[2] ← ISPACE[2]; FOR I ← 1 STEP 1 UNTIL 2 DO BEGIN PRDUCT[I] ← XYU[ISC   369
] × XYL[I]; LL[I] := 0; IF IJ[I] < 0 THEN GO TO L350; IF PRDUCT[I] < 0 TSC   370
HEN GO TO L360 ELSE IF PRDUCT[I] > 0 THEN GO TO L350; IF XYL[I] ≠ 0 THENSC   371
 GO TO L360; L360: GO TO SW1[I]; ; L350: LL[I] ● ISPACE[I]; L370: IF LL[SC   372
1] > (LU[1] + IWIDE + 2) THEN GO TO L420 ELSE LL[1] ← LU[1] + IWIDE + 2;SC   373
 L420: ML[I] ← LL[I] + MTL[I]; MU[I]← LU[I] + MTU[I]; ITT[I] ← 1023 - MLSC   374
[I] - MU[I] END; IF NM[1] < NM[2] THEN BEGIN IT ← NM[1]; GO TO L460 END SC   375
ELSE IT ← NM[2]; L460: IF IT ≥ 0 THEN GO TO L660 ELSE IF ITT[1] < ITT[2]SC   376
 THEN BEGIN IT ← 1023 - ITT[1]; GO TO L480 END ELSE IT ← 1023 - ITT[2]; SC   377
L480: MU[1] ← IT - ML[1]; MU[2] ← IT - ML[2]; COMMENT.....ERROR TESTS ANSC   378
D SCALE; L660: IF K[1] ≠ 0 THEN GO TO L668 ELSE ERRLNV(XYL[1],XYU[1],ML[SC   379
1], MU[1],DXY[1]); GO TO L670; L668: ERRNLV(XYL[1],XYU[1],ML[1],MU[1],DXSC   380
Y[1]); L670: AML ← ML[1]; AMU ← MU[1]; XSCALV(XYL[1],XYU[1],AML,AMU,0,0,SC   381
0,0); IF K[2] ≠ 0 THEN GO TO L690 ELSE ERRLNV(XYL[2],XYU[2],ML[2], MU[2]SC   382
,DXY[2]); GO TO L700; L690: ERRNLV(XYL[2],XYU[2],ML[2],MU[2],DXY[2]); L7SC   383
00: AML ← ML[2]; AMU ← MU[2]; YSCALV(XYL[2],XYU[2],AML,AMU,0,0,0,0); LRESC   384
FR[1] ← MTL[1] + IWD2 + 3; LREFR[2] ← MTL[2] + IHD2 + 3; FOR I ← 1 STEP SC   385
1 UNTIL 2 DO BEGIN PRDUCT[I] ← XYL[I] × XYU[I]; IF IJ[I] < 0 THEN GO TO SC   386
L790 ELSE IF PRDUCT[I] > 0 THEN GO TO L790 ELSE GO TO SW2[I]; L752: XSCASC   387
LV(0,0,0,0,1, 0,ITT[1],0); ITT[1] ← ITT[1] - ISPACE[1] + IWD2; GO TO L75SC   388
8; L756: YSCALV(0,0,0,0,1, 0,ITT[2],0); ITT[2] ● ITT[2] - ISPACE[2] + IHSC   389
D2; L758: IF (ITT[I] - MTL[I]) ≥ 0 THEN GO TO L780 ELSE IJ[I] ← -(IJ[I])SC   390
; GO TO L260; L780: LREFR[I] ← ITT[I] + 2; L790: END; MUH ← 1023 - MU[1]SC   391
; MUV ← 1023 - MU[2]; IF YMIN ≥ 0 THEN GO TO L810 ELSE LREFR[1] ← LREFR[SC   392
1] + IWIDE; L810: IF K[1] ≠ 0 THEN GO TO L850 ELSE LINRV(1,LREFR[2],ML[2SC   393
],MUV, XYL[1],XYU[1],DXY[1],NM[1],IJ[1],NXY[1],IWIDE); GO TO L870; L850:SC   394
 NONLNV(1,LREFR[2],ML[2],MUV,XYL[1],XYU[1],DXY[1],NM[1],IJ[1], NXY[1],IWSC   395
IDE); L870: IF K[2] ≠ 0 THEN GO TO L910 ELSE LINRV(2,LREFR[1],ML[1],MUH,SC   396
 XYL[2],XYU[2],DXY[2],NM[2],IJ[2],NXY[2],IHIGH); GO TO L930; L910: NONLNSC   397
V(2,LREFR[1],ML[1],MUH,XYL[2],XYU[2],DXY[2],NM[2],IJ[2], NXY[2],IHIGH); SC   398
L930: END; PROCEDURE QUIK3LP(L,XL,XR,YB,YT,ISYM,BCDX,BCDY,NP,X,Y); VALUESC   399
 L,XL,XR,YB,YT,ISYM,NP; COMMENT.....QUIK3L WILL PLOT 1,2 OR 3 GRAPHS/FRASC   400
```

```
ME ON SC-4020 AND CONNECTS PTS.; REAL ARRAY X,Y,BCDX,BCDY[1]; INTEGER L,SC   401
 NP; ALPHA ISYM; REAL XL,XR,YB,YT; BEGIN INTEGER L1,NCX,DCX,DCY,INCRY,ICSC   402
Y,IX,IY,IY1,NCY,N,NX,IER,M,J, NY,K,I,NX0,NY0,NX1,NY1,A,B; LABEL L100,L11SC   403
0,L140,L142,L144,L150,L200,L201,L210,L220,L271, L146,START; ALPHA ARRAY SC   404
ASYM[1:1]; REAL DX,DY; SWITCH SW3 := L142, L144, L146; START: ASYM[1] := SC   405
ISYM; IF L = 0 THEN GO TO L200 ELSE IF L > 0 THEN GO TO L100 ELSE L1 := 1SC   406
; GO TO L110; L100: L1 := 2; L110: NCX := NBLANK(BCDX,12); NCY := NBLANK(BCSC   407
DY,12); DCX := DCY := 10; INCRY := -14; L140: MARGIN(L,ICY,A,B); IX := 524 -SC   408
(4 * NCX); IY := ICY +(7 * NCY); GO TO SW3[L]; L142: IY1 := 0; GO TO L150;SC   409
 L144: IY1 := ICY - 253; GO TO L150; L146: IY1 := ICY - 169; L150: CAMRAVSC   410
(35); DXDYV(1,XL,XR,DX,N,I,NX,DCX,IER,A,B); DXDYV(2,YB,YT,DY,M,J,NY,DCY,SC   411
IER,A,B); GRID1V(L1,XL,XR,YB,YT,DX,DY,N,M,I,J,NX,NY,A,B); PRINTV(NCX,BCDSC   412
X,IX,IY1); APRNTV(0,INCRY,NCY,BCDY,0,IY); L200: IF NP ≥ 0 THEN BEGIN APLSC   413
OTV(NP,X,Y,1,1,1,ASYM,IER); GO TO L271; END ELSE NP := ABS(NP); L201: FOSC   414
R K := 1 STEP 1 UNTIL NP DO BEGIN XSCALV(0,0,0,0,1,X[K],NX1,0); YSCALV(0,SC   415
0,0,0,1,Y[K],NY1,0); IF (K-1) ≤ 0 THEN GO TO L220 ELSE L210: LINEV(NX0,NSC   416
Y0,NX1,NY1); L220: PLOTV(NX1,NY1,ISYM,0);NX0 := NX1; NY0 := NY1 END; L271:SC   417

 END; PROCEDURE QUIK3V (L,ISYM,BCDX,BCDY,NP,X,Y); VALUE ISYM,L,NP; REAL SC   418
ARRAY X,Y,BCDX,BCDY[1]; INTEGER L,NP; ALPHA ISYM; COMMENT.....COMPUTES XSC   419
,Y,MIN, AND MAX AND PLOTS ON SC-4020 L = NO. GRAPHS/FRAME.(1,2,OR 3). IFSC   420
 L IS NEG, FRAME WILL BE ADVANCED AND NP POINTS WILL BE PLOTTED ON GRAPHSC   421
 NO. 1 IF L IS POS, NP POINTS WILL BE PLOTTED ON GRAPH NO. 2 AND 3 ISYM SC   422
= PLOTTING SYMBOL NP = NO. PTS TO BE PLOTTED BCDX,BCDY = ALPHANUM CHARACSC   423
TERS FOR X AND Y LABELS IF NP IS NEG, PTS WILL BE CONNECTED WITH STRAIGHSC   424
T LINE; BEGIN INTEGER K; LABEL L60,L100; REAL XL,XR,YB,YT,XRB,XLB,YTB,YBSC   425
B; XL := X[1]; XR := X[1]; YB := Y[1]; YT := Y[1]; FOR K := 2 STEP 1 UNTIL ABSC   426
S(NP) DO BEGIN IF X[K] < XL THEN XL := X[K]; IF X[K] > XR THEN XR := X[K];SC   427
 IF Y[K] < YB THEN YB := Y[K]; IF Y[K] > YT THEN YT := Y[K] END; SCRND(XR,SC   428
XL,XRB,XLB); XR := XRB; XL := XLB; SCRND(YT,YB,YTB,YBB); YT := YTB; YB := YBSC   429
B; IF NP = 0 THEN GO TO L100; QUIK3LP(L,XL,XR,YB,YT,ISYM,BCDX,BCDY,NP,X,SC   430
Y); L100: END; PROCEDURE IDENT(ADARRAY,DATE); VALUE DATE; ALPHA DATE; COSC   431
MMENT.....WILL WRITE ID FRAME OR EOJ FRAME IN BLOCK LETTERS; ALPHA ARRAYSC   432
 ADARRAY[1]; COMMENT.....PRESET IS INCORPORATED INTO THIS ROUTINE; BEGINSC   433
 LABEL REPET, KAPUT1,KAPUT2,SWCT1,SWCT2,SWCT3,SWCT4,CARDPRO, ID,KAPUT3,BSC   434
```

```
LOP,GO1; SWITCH SWPRO := SWCT1,SWCT2,SWCT3,SWCT4; INTEGER J,M,N,X,Y,I,TESC   435
MP1,TEMP2,CT1; INTEGER ARRAY CTCOL,FIELDCT[1:4]; INTEGER ARRAY TABLE[1:1SC   436
27]; DEFINE JOBID=THISISJOBID#,Z=TOMJIMJOEJOHN#; DEFINE IDARRAY=B222IDARSC   437
RAY#, A=DRATTYA#; ALPHA ARRAY CDAREA[1:46], JOBNO[1:28]; WRITE4020SWITCHSC   438
1=0; XYZWPAGECT:=0; COMMENT....TABLES FOR DRAW VECTOR IDRECORDS.; FILL TSC   439
ABLE[*] WITH OCT0640020640120, OCT0600040576100, OCT0700020600200, OCT06SC   440
60013640120, OCT0660040460100, OCT0740070120130, OCT0740010600200, OCT07SC   441
40010600100, OCT0740070060100, OCT0740010450130, OCT0740070050154, OCT06SC   442
20020120100, OCT0740010600150, OCT0600050576100, OCT0740070200100, OCT06SC   443
00010460100, OCT0740010450130, OCT0740070050154, OCT0600010576100, OCT07SC   444
40010600200, OCT0600070660200, OCT0740070200150, OCT0740010600100, OCT06SC   445
60070176100, OCT0640040600140, OCT0600070776200, OCT0740070200100, OCT06SC   446
00010460100, OCT0740010600130, OCT0740010600130, OCT0600070440130, OCT07SC   447
40070200150, OCT0600010640150, OCT0700020440160, OCT0660060140200, OCT06SC   448
20030720260, OCT0640040260210, OCT0640030600160, OCT0600050440160, OCT06SC   449
40050200200, OCT0600030640200, OCT0660010776200, OCT0660040576100, OCT07SC   450
00020600150, OCT0740010440100, OCT0740070040120, OCT0740010440140, OCT07SC   451
40070040160, OCT0600010776200, OCT0740070200100, OCT0600010576100, OCT07SC   452
40010600200, OCT0600070776200, OCT0740010500100, OCT0740070100140, OCT06SC   453
00010776200, OCT0740070200100, OCT0600010576100, OCT0720010600140, OCT07SC   454
40010600200, OCT0740070200100, OCT0600010576100, OCT0740010600200, OCT06SC   455
00070640200, OCT0640070200160, OCT0600010576100, OCT0740010600140, OCT06SC   456
00070576100, OCT0700020600100, OCT0600040576100, OCT0700020600200, OCT07SC   457
00030600100, OCT0600050576100, OCT0700050200200, OCT0600010640200, OCT06SC   458
00010576100, OCT0720060100100, OCT0740010500140, OCT0600010576100, OCT07SC   459
40010600200, OCT0600070640200, OCT0600070740160, OCT0740070200100, OCT06SC   460
40040500160, OCT0600010776200, OCT0660010460100, OCT0660040660130, OCT06SC   461
00070576100, OCT0600010776200, OCT0740010576100, OCT0600070776200, OCT06SC   462
00010776200, OCT0740010600100, OCT0600070500100, OCT0740070200140, OCT07SC   463
40010500140, OCT0740070200100, OCT0600010500100, OCT0740010600140, OCT06SC   464
00070500140, OCT0740070200200, OCT0600010620110, OCT0740010600100, OCT06SC   465

00070420100, OCT0600040576100, OCT0600010576100, OCT0740010600200, OCT06SC   466
00070776200, OCT0740070200100, OCT0700020600140, OCT0660010576100, OCT06SC   467
60040776200, OCT0620010576100, OCT0640020660200, OCT0640040460150, OCT06SC   468
```

```
20060776200, OCT0740010576100, OCT0740010776200, OCT0740010600100, OCT07SC   469
40010600200, OCT0660010460100, OCT0660070060100, OCT0600040520130, OCT07SC   470
40010600100, OCT0740070176100, OCT0740010600200, OCT0700020600140; IF DASC   471
TF ≠ "999999" THEN GO TO ID ELSE RESETV; FILL JOBID[*] WITH "E","N","D",SC   472
" ","J","O","B"," "; COMMENT DELETE THE FOLLOWING FEW WHEN FILL IS OK; JSC   473
OBID[9]:=IDARRAY[1].[12:6]; JOBID[10]:=IDARRAY[1].[18:6]; JOBID[11]:=IDASC   474
RRAY[1].[24:6]; JOBID[12]:=IDARRAY[1].[30:6]; JOBID[13]:=IDARRAY[1].[36:SC   475
6]; JOBID[14]:=IDARRAY[1].[42:6]; JOBID[15]:=IDARRAY[2].[12:6]; JOBID[16SC   476
]:=IDARRAY[2].[18:6]; X := 0; Y := 128; FOR I := 1 STEP 1 UNTIL 16 DO BESC   477
GIN IF JOBID[I] = " " THEN GO TO KAPUT3 ELSE TEMP1 := B5CODE(JOBID[I]); SC   478
FOR CT1 := 0 STEP 1 UNTIL (TEMP1.[39:3] - 1) DO BEGIN TEMP2 := TABLE[TEMSC   479
P1.[30:9] + CT1] + X + Y; PLOTP(0,TEMP2); PLOTP(0,TEMP2) END; KAPUT3: X.SC   480
[20:10] := X.[20:10] + 64 END; FOR I := 1 STEP 1 UNTIL 5 DO RESETV; COMMSC   481
ENT.....INSERT TEST4020SWITCH HERE ********************************; LOCSC   482
K(FT4020,SAVE); GO TO BLOP; ID: FILL JOBNO[*] WITH "J","O","B"," ","N","SC   483
O",".","D","A","T","E", "S","E","N","D"," ","T","O","P","R","O","G","R",SC   484
 "A","M","M","E","R"; FILL JOBID[*] WITH "J","O","B"," ","N","O",".","," "SC   485
; FOR I:=1 STEP 1 UNTIL 7 DO IDARRAY[I] := ADARRAY[I]; Z := 1; SETSCALE(SC   486
0,0,0); FILL FIELDCT[*] WITH 8,46,23,38; FILL A[*] WITH OCT0062400, OCTOSC   487
001301, OCT0004402, OCT0010403, OCT0014304, OCT0017405, OCT0023406, OCTOSC   488
027307, OCT0165410, OCT0032411, OCT0000012, OCT0000055, OCT0000037, OCTOSC   489
000015, OCT0000016,OCT0000017, OCT0000020, OCT0052321, OCT0055522, OCT00SC   490
62323, OCT0066324, OCT0071425, OCT0071326, OCT0075527, OCT0102330, OCT01SC   491
05331, OCT0046433, OCT0000035, OCT0000075, OCT0000074, OCT0000036, OCT00SC   492
00077, OCT0000052, OCT0110441, OCT0114342, OCT0117343, OCT0125444, OCT01SC   493
31345, OCT0152546, OCT0134447, OCT0117650, OCT0134551, OCT0000053, OCT00SC   494
00054, OCT0036440, OCT0000034, OCT0000056, OCT0000057, OCT0000060, OCT00SC   495
00061, OCT0141562, OCT0146463, OCT0152364, OCT0157265, OCT0161466, OCT01SC   496
65267, OCT0171370, OCT0174471, OCT0042473, OCT0000032, OCT0000072, OCT00SC   497
00013, OCT0000076, OCT0000014; FILL CTCOL[*] WITH 1,39,9,24; IDARRAY[8] SC   498
:= DATE; J := 1; N := 0; FOR I := 1 STEP 1 UNTIL 8 DO BEGIN IF I = 7 THESC   499
N BEGIN N := 4; GO TO GO1 END; CDAREA[J] := IDARRAY[I].[12:6]; CDAREA[J+SC   500
1] := IDARRAY[I].[18:6]; IF I = 8 THEN BEGIN J := J + 1; CDAREA[J+1] := SC   501
```

```
"-" END; CDAREA[J+2] := IDARRAY[I].[24:6]; CDAREA[J+3] :=IDARRAY[I].[30:SC    502
6]; IF I = 8 THEN BEGIN J := J + 1; CDAREA[J+3] := "-" END; GO1: CDAREA[SC    503
J-N+4] := IDARRAY[I].[36:6]; CDAREA[J-N+5] := IDARRAY[I].[42:6]; IF I = SC    504
7 THEN J := J + 2 ELSE J := J + 6; N := 0 END; FOR I := 9 STEP 1 UNTIL 1SC    505
6 DO JOBID[I] := CDAREA[I-8]; J := M := 1; N := 6; RESETV; Y := 64; GRAPSC    506
HCTPERPAGE := 1; REPET: X:=0; FOR I :=M STEP 1 UNTIL (N+M) DO BEGIN IF JSC    507
OBNO[I] = " " THEN GO TO KAPUT1 ELSE TEMP1 := B5CODE(JOBNO[I]); FOR CT1 SC    508
:=0 STEP 1 UNTIL (TEMP1.[39:3] - 1) DO BEGIN TEMP2:= TABLE[TEMP1.[30:9] SC    509
+ CT1] + X + Y; PLOTP(0,TEMP2); PLOTP(0,TEMP2); END; KAPUT1: X.[20:10]:=SC    510
 X.[20:10] + 64 END; GO TO SWPRO[J]; SWCT1: X.[20:10] := X.[20:10] + 64;SC    511
 N := 3; M := 8; GO TO CARDPRO; SWCT2: X.[20:10] := X.[20:10] + 128; N :SC    512
= 6; M := 12; GO TO CARDPRO; SWCT3: X.[20:10] := 64; Y := Y + 128; N := SC    513
9; M := 19; GO TO CARDPRO; SWCT4: X.[20:10] := 64; Y := Y + 128; CARDPROSC    514
: FOR I := CTCOL[J] STEP 1 UNTIL FIELDCT[J] DO BEGIN IF CDAREA[I] = " " SC    515
THEN GO TO KAPUT2 ELSE TEMP1 := B5CODE(CDAREA[I]); FOR CT1 := 0 STEP 1 USC    516
NTIL (TEMP1.[39:3] - 1) DO BEGIN TEMP2:= TABLE[TEMP1.[30:9] + CT1] + X +SC    517
 Y; PLOTP(0,TEMP2) END; KAPUT2: X.[20:10]:= X.[20:10] + 64 END; Y:= Y + SC    518
128; J:= J +1; IF J# 5 THEN GO TO REPET ELSE RESETV; BLOP: END; PROCEDURSC    519
E ENDJOB; COMMENT.....DUMMY WILL CALL IDENT TO WRITE END OF JOB FRAME; BSC    520
EGIN IDENT(B222IDARRAY,"999999") END; COMMENT LAST CARD OF SC4020 ROUTINSC    521
ES IS SC 8990;                                                          S6    522

COMMENT FOLLOWING ARE THE DECLARATIONS FOR THE PROGRAM ;                      002
REAL  SUM6TX,SUM6TY,SUM6XT1,SUM6YT1,SUM6DXT1,SUM6DYT1,SUM3X1,SUM3Y1,         003
      HTI,SUM4DX1,SUM4DY1,IHTI,Y1SQ,FVALXT,FVALYT,DFVALXT,DFVALYT,           004
      SUM3X,SUM3Y,DTI,IDTIM1,SUM4X,SUM4Y,YSQ,SUM5FT,SUM5GT,IIM1DTIM2,        005
      YOSQ,UNUN,VNUN,XNUNP1,YNUNP1,EPS1,EPS2,TMAX,DYHOLD,DJ,J1,RMIN,         006
      RMAX,SMIN,SMAX,DXA,DYA,TTEST,PER,DIR,TSTART,XSTART,YSTART,DXSTART,     007
      DYSTART,HSTART,MU,MUPM,C1,C2,EPSM,M1,M2,M3,M4,M5,M6,M7,M8,D1,D2,       008
      TERM,TERM1,TERM2,TERM3,TERM4,X0,DX0,Y0,DY0,XT,YT,DXT,DYT,DT,X,Y,       009
      DX,DY,DEN1,DEN2,F,G,FT,GT,H,TX,TY,EPS,EPSX,EPSY,ERRX,ERRY,XT1,YT1,     010
      DXT1,DYT1,X1,Y1,DX1,DY1,T,R,S,J ;                                       011
INTEGER I,NMI,NM1MI,DIR1,DIR2,LM1,NP,CC,M,MP1,MP2,MP3,MP4,MP5,N,NP1,         012
        NM1,LAMDA,DUUM,SAME ;                                                 013
```

```
BOOLEAN ITER,PRITR,PLT,DUB ;                                                014
ALPHA DATE ;                                                                015
LABEL PRTEST,PRNT,SWAP,HALVE,QUIT,INTERP,STEST,EOF,AHEAD,TEST,ITERATE,      016
      START,TSTEP,LOOP ;                                                    017
ALPHA ARRAY FRAMEID[0:8],ABSCISSA,ORDINATE[0:12] ;                           018
REAL ARRAY PLOTX,PLOTY[0:400],ALF[0:4],FCON[0:4,0:4],DFCON[0:4,0:4],        019
           C[0:4],CPM[0:4],CHAT[0:4],XNU[0:15],YNU[0:15],RNU[0:15],         020
           SNU[0:15],UNU[0:15],VNU[0:15],K[0:4],L[0:4],CDIF[0:4],           021
           SIG[0:15] ;                                                      022
 FILE OUT PRINT 1(3,15) ;                                                   023
 FILE IN PCARD(1,10) ;                                                      024
 FORMAT IN PFMT(E18.11  /  5E15.8 /  E15.8 /   3E15.8 /   E15.8 /   I3     025
                /  E15.8,2I2 /   4E18.11 /  3L5) ;                          026

FORMAT OUT LFMT(X3," T = ",E17.10," X = ",E17.10," Y = ",E17.10,            027
                " XD = ",E17.10," YD = ",E17.10//X3,"DT = ",E17.10,         028
                "  J = ",E17.10," DJ = ",E17.10," RE = ",E17.10,            029
                " RM = ",E17.10///);                                        030
 FORMAT IN AFMT(AA  /  8A6) ;                                               031
LIST PLINE(T,X1,Y1,DX1,DY1,H,J,DJ,R,S);                                     032
LIST PLINE1(T,X0,Y0,DX0,DY0,H,J,DJ,R,S);                                    033
 LIST PLIST(MU,RMAX,RMIN,SMAX,SMIN,TMAX,TSTART,EPS,EPS1,EPS2,HSTART,        034
            M,PER,DIR1,DIR2,XSTART,YSTART,DXSTART,DYSTART,ITER,PRITR,       035
            PLT) ;                                                          036
 LIST ALIST(DATE,FOR DUUM ← 1 STEP 1 UNTIL 8 DO FRAMEID[DUUM]) ;            037
COMMENT FOLLOWING IS THE MAIN BODY OF THE PROGRAM ;                          038
FILL ABSCISSA[*] WITH "X     ", "      ", "      ", "      ",               039
                      "      ", "      ", "      ", "      ",               040
                      "      ", "      ", "      ", "      ";              041
FILL ORDINATE[*] WITH "Y     ", "      ", "      ", "      ",               042
                      "      ", "      ", "      ", "      ",               043
                      "      ", "      ", "      ", "      ";              044
  START: READ(PCARD,PFMT,PLIST)[EOF] ;                                      045
        IF PLT THEN                                                         046
        READ(PCARD,AFMT,ALIST)[EOF] ;                                       047
```

```
IF ITER THEN                                                    048
BEGIN                                                           049
  TTEST ← (0.4)×PER ;                                           050
  SAME  ← DIR1                                                  051
END                                                             052
ELSE                                                            053
BEGIN                                                           054
  TTEST ← (0.9)×PER ;                                           055
  SAME ← DIR2 ;                                                 056
END ;                                                           057
DXA ← 0.00005 ;                                                 058
DYA ← DYSTART + 0.00005 ;                                       059
NP ← 1 ;                                                        060

CC ← 1 ;                                                        061
WRITE(PRINT[PAGE]) ;                                            062
MP1 ← M + 1 ;                                                   063
MP2 ← M + 2 ;                                                   064
MP3 ← M + 3 ;                                                   065
MP4 ← M + 4 ;                                                   066
MP5 ← M + 5 ;                                                   067
MUPM ← 1 - MU ;                                                 068
C1 ← 2×MU ;                                                     069
C2 ← 2×MUPM ;                                                   070
EPSM ← (0.5)*MP5 ;                                              071

M1 ← MP2/MP4 ;                                                  072
M2 ← M1*MP1 ;                                                   073
M3 ← MP4/MP2 ;                                                  074
M4 ● M3*MP1 ;                                                   075
M5 ← MP2/MP5 ;                                                  076
M6 ← M5*MP1 ;                                                   077
M7 ← MP5*2 ;                                                    078
M8 ← MP2×MP3 ;                                                  079
ALF[1] ● 1 ;                                                    080
```

```
          ALF[2] ← M1 ⍝                                                    081
          ALF[3] ← 1 ⍝                                                     082
          ALF[4] ← M5 ⍝                                                    083
          FCON[1;1] ← (2×M2)/(MP4*2) ⍝                                     084
          FCON[2;1] ← 0 ⍝                                                  085
          FCON[2;2] ← 0 ⍝                                                  086
          FCON[3;1] ← 0 ⍝                                                  087
          FCON[3;2] ← 0 ⍝                                                  088
          FCON[3;3] ← ((1.5)×M6)/M7 ⍝                                      089
          DFCON[1;1] ← M2/MP4 ⍝                                            090
          DFCON[2;1] ← -(1/MP2) ⍝                                          091
          DFCON[2;2] ← (2×M4)/MP2 ⍝                                        092
          DFCON[3;1] ← ((0.5)×M6)/MP5 ⍝                                    093
          DFCON[3;2] ← ((0.25)×(5×M + 16)×((MP4/MP5)*MP1))/M7 ⍝            094
          DFCON[3;3] ← ((-0.75)×MP2×M6)/M7 ⍝                               095
          C[1] ← 0 ⍝                                                       096
          C[2] ← M4/M8 ⍝                                                   097
          C[3] ← 0 ⍝                                                       098
          CPM[1] ← 0 ⍝                                                     099
          CPM[2] ← ((0.5)×MP4×M4)/M8 ⍝                                     100
          CPM[3] ← (0.5)/MP3 ⍝                                             101
          CHAT[1] ← 0 ⍝                                                    102
          CHAT[2] ← 0 ⍝                                                    103
          CHAT[3] ← (0.333333333333)/(MP3×MP4) ⍝                           104
          CHAT[4] ← ((0.66666666667)×MP5×((MP5/MP2)*MP1))/(M8×MP4) ⍝       105
ITERATE:X0 ← XSTART ⍝                                                      106
          Y0 ← YSTART ⍝                                                    107
          DX0 ← DXSTART ⍝                                                  108
          DY0 ← DYSTART ⍝                                                  109
          H ← HSTART ⍝                                                     110
          T ← TSTART ⍝                                                     111
          DUB ← TRUE ⍝                                                     112
          R ← X0 + MU ⍝                                                    113
          S ← X0 - MUPM ⍝                                                  114
```

```
        J ←0.0 ;                                                    115
        J1 ← 0.0 ;                                                  116
        DJ ←0.0 ;                                                   117
        IF ITER THEN GO TO PRTEST ;                                 118

        IF PLT THEN                                                 119
        BEGIN                                                       120

          PLOTX[NP] ← XO ;                                          121

          PLOTY[NP] ← YO ;                                          122
          NP ← NP + 1                                               123
        END ;                                                       124
        GO TO PRNT ;                                                125
PRTEST:IF PRITR THEN                                                126
        PRNT:BEGIN                                                  127
               IF CC = 11 THEN                                      128
               BEGIN                                                129
                 CC ← 1 ;                                           130
                 WRITE(PRINT[PAGE])                                 131
               END ;                                                132
               WRITE(PRINT,LFMT,PLINE1)                             133
             END ;                                                  134
TSTEP: XNU[0] ← XO ;                                                135
        XNU[1] ● DXO ;                                              136
        YNU[0] ← YO ;                                               137
        YNU[1] ← DYO ;                                              138
        YOSQ ● YO*2 ;                                               139
        RNU[0] ← SQRT((XO + MU)*2 + YOSQ) ;                         140
        SNU[0] ← SQRT((XO - MUPM)*2 + YOSQ) ;                       141
        UNU[0] ← MUPM / (RNU[0]*3) ;                                142

        VNU[0] ● MU / (SNU[0]*3) ;                                  143
        FOR N ← 1 STEP 1 UNTIL MP2 DO                               144
        BEGIN                                                       145
          NP1 ← N + 1 ;                                             146
          NM1 ← N - 1 ;                                             147
```

```
D1 ← 2×N ;                                                          148
D2 ← N×NP1 ;                                                        149
TERM ← 0 ;                                                          150
FOR I ← 0 STEP 1 UNTIL N DO                                         151
BEGIN                                                               152
  NMI ← N - I ;                                                     153
  TERM ← TERM + (XNU[I]× XNU[NMI]) + (YNU[I]×YNU[NMI]) ;            154
END ;                                                               155
TERM1 ← 0 ;                                                         156
TERM2 ← 0 ;                                                         157
TERM3 ← 0 ;                                                         158
TERM4 ← 0 ;                                                         159
IF N ≠ 1 THEN                                                       160
BEGIN                                                               161
  FOR I ← 1 STEP 1 UNTIL NM1 DO                                     162
  BEGIN                                                             163
    NMI ← N - I ;                                                   164
    TERM1 ← TERM1 + (RNU[I]×RNU[NMI]) ;                             165
    TERM2 ← TERM2 + (SNU[I]×SNU[NMI]) ;                             166
    TERM3 ← TERM3 + (I×UNU[I]×RNU[NMI]) ;                           167
    TERM4 ← TERM4 + (I×VNU[I]×SNU[NMI]) ;                           168
  END ;                                                             169
END ;                                                               170
RNU[N] ← (TERM + C1×XNU[N] - TERM1)/(2×RNU[0]) ;                    171
SNU[N] ← (TERM - C2×XNU[N] - TERM2)/(2×SNU[0]) ;                    172
UNUN ← 0 ;                                                          173
VNUN ← 0 ;                                                          174
FOR I ← 1 STEP 1 UNTIL N DO                                         175
BEGIN                                                               176
  NMI ← N - I ;                                                     177
  UNUN ← UNUN + (I×RNU[I]×UNU[NMI]) ;                               178
  VNUN ← VNUN + (I×SNU[I]×VNU[NMI]) ;                               179
END ;                                                               180
```

```
          UNU[N] ← (((-3)×UNUN) - TERM3) / (N×RNU[0]) ;                    181
          VNU[N] ← (((-3)×VNUN) - TERM4) / (N×SNU[0]) ;                    182
          IF N = MP2 THEN GO TO LOOP ;                                     183
          FOR I ← 0 STEP 1 UNTIL NM1 DO                                    184
          SIG[I] ← UNU[I] + VNU[I] ;                                       185
          XNUNP1← 0 ;                                                      186
          YNUNP1 ← 0 ;                                                     187
          FOR I ← 0 STEP 1 UNTIL NM1 DO                                    188
          BEGIN                                                            189
          NM1MI ← NM1 - I ;                                                190
          XNUNP1 ← XNUNP1 + (SIG[I]×XNU[NM1MI]) ;                          191
          YNUNP1 ← YNUNP1 + (SIG[I]×YNU[NM1MI]) ;                          192
          END ;                                                            193
          XNU[NP1] ← (XNU[NM1] + (D1×YNU[N]) - (MU×UNU[NM1])               194
            + (MUPM×VNU[NM1]) - XNUNP1) / D2 ;                             195
          YNU[NP1] ← (YNU[NM1] - (D1×XNU[N]) - YNUNP1) / D2  ;             196
        END ;                                                              197
LOOP:   FOR LAMDA ← 1 STEP 1 UNTIL 4 DO                                    198
        BEGIN                                                              199
          LM1 ← LAMDA - 1 ;                                                200
          FVALXT ← 0 ;                                                     201
          FVALYT ← 0 ;                                                     202
          FOR I ← I STEP 1 UNTIL LM1 DO                                    203
          BEGIN                                                            204
            FVALXT ← FVALXT + (FCON[LM1,I]×K[I]) ;                         205
            FVALYT ← FVALYT + (FCON[LM1,I]×L[I]) ;                         206
          END ;                                                            207
          XT ← X0 + (H×FVALXT) ;                                           208
          YT ← Y0 + (H×FVALYT) ;                                           209
          DFVALXT ← 0 ;                                                    210
          DFVALYT ← 0 ;                                                    211
          FOR I ← 1 STEP 1 UNTIL LM1 DO                                    212
```

```
BEGIN                                                                  213
  DFVALXT ← DFVALXT + (DFCON[LM1,I]×K[I]) ;                            214
  DFVALYT ← DFVALYT + (DFCON[LM1,I]×L[I]) ;                            215
END ;                                                                  216
DXT ← H×DFVALXT ;                                                      217
DYT ← H×DFVALYT ;                                                      218
DT ← ALF[LAMDA]×H ;                                                    219
SUM3X ← 0 ;                                                            220
SUM3Y ← 0 ;                                                            221
FOR I ← 1 STEP 1 UNTIL MP2 DO                                          222
BEGIN                                                                  223
  DTI ← DT*I ;                                                         224
  SUM3X ← SUM3X + (XNU[I]×DTI) ;                                       225
  SUM3Y ← SUM3Y + (YNU[I]×DTI) ;                                       226
END ;                                                                  227
X ← XT + SUM3X ;                                                       228
Y ← YT + SUM3Y ;                                                       229
SUM4X ← 0 ;                                                            230
SUM4Y ← 0 ;                                                            231
FOR I ← 1 STEP 1 UNTIL MP2 DO                                          232
BEGIN                                                                  233
  IDTIM1 ← I×(DT*(I - 1)) ;                                            234
  SUM4X ← SUM4X + (IDTIM1×XNU[I]) ;                                    235
  SUM4Y ← SUM4Y + (IDTIM1×YNU[I]) ;                                    236
END ;                                                                  237
DX ← DXT + SUM4X ;                                                     238
DY ← DYT + SUM4Y ;                                                     239
YSQ ← Y ↑ 2 ;                                                          240
DEN1 ← ((X+MU)↑2 + YSQ)↑1.5 ;                                          241
DEN2 ← ((X-MUPM)↑2 + YSQ)↑1.5 ;                                        242
F ← X + (2×DY) - ((MUPM×(X + MU))/DEN1)                                243
  -((MU×(X - MUPM))/DEN2) ;                                            244
G ← Y - (2×DX) - (Y×((MUPM/DEN1) + (MU/DEN2))) ;                       245
SUM5FT ← 0 ;                                                           246
```

```
            SUM5GT ← 0 ;                                                    247
            FOR I ← 2 STEP 1 UNTIL MP2 DO                                   248
            BEGIN                                                           249
              IIM1DTIM2 ← I×(I - 1)×(DT*(I - 2)) ;                          250
              SUM5FT ← SUM5FT + (IIM1DTIM2×XNU[I]) ;                        251
              SUM5GT ← SUM5GT + (IIM1DTIM2×YNU[I]) ;                        252
            END ;                                                           253
            FT ← F - SUM5FT ;                                               254
            GT ← G - SUM5GT ;                                               255
            K[LAMDA] ← FT×H ;                                               256
            L[LAMDA] ← GT×H;                                                257
          END ;                                                             258

          FOR I ← 1 STEP 1 UNTIL 4 DO                                       259
          CDIF[I] ← C[I] - CHAT[I] ;                                        260
          SUM6TX ← 0 ;                                                      261
          SUM6TY ← 0 ;                                                      262
          FOR I ← 1 STEP 1 UNTIL 4 DO                                       263
          BEGIN                                                             264
            SUM6TX ← SUM6TX + (CDIF[I]×K[I]) ;                              265
            SUM6TY ← SUM6TY + (CDIF[I]×L[I]) ;                              266
          END ;                                                             267
          TX ← ABS(H×SUM6TX) ;                                              268
          TY ← ABS(H×SUM6TY) ;                                              269

          EPSX ← ABS(X0×EPS) ;                                              270
          EPSY ← ABS(Y0×EPS) ;                                              271
          IF EPSX = 0 THEN ERRX ← EPS ELSE                                  272
          ERRX ← TX/EPSX ;                                                  273
          IF EPSY = 0 THEN ERRY ← EPS ELSE                                  274
          ERRY ← TY/EPSY ;                                                  275
          IF (ERRX > 1) OR (ERRY > 1) THEN                                  276
HALVE:    BEGIN                                                             277
            H← (0.5)×H ;                                                    278
            GO TO LOOP                                                      279
```

```
END ;                                                          280

SUM6XT1 ← 0 ;                                                  281

SUM6YT1 ← 0 ;                                                  282

FOR I ← 1 STEP 1 UNTIL 3 DO                                    283

BEGIN                                                          284

  SUM6XT1 ← SUM6XT1 + (C[I]×K[I]) ;                            285

  SUM6YT1 ← SUM6YT1 + (C[I]×L[I]) ;                            286

END ;                                                          287

XT1 ← X0 + (H×SUM6XT1) ;                                       288

YT1 ← Y0 + (H×SUM6YT1) ;                                       289

SUM6DXT1 ← 0 ;                                                 290

SUM6DYT1 ← 0 ;                                                 291

FOR I ← 1 STEP 1 UNTIL 3 DO                                    292

BEGIN                                                          293

  SUM6DXT1 ← SUM6DXT1 + (CPM[I]×K[I]) ;                        294

  SUM6DYT1 ← SUM6DYT1 + (CPM[I]×L[I]) ;                        295

END ;                                                          296

DXT1 ← SUM6DXT1 ;                                              297

DYT1 ← SUM6DYT1 ;                                              298

SUM3X1 ← 0 ;                                                   299

SUM3Y1 ← 0 ;                                                   300

FOR I ← 1 STEP 1 UNTIL MP2 DO                                  301

BEGIN                                                          302

  HTI ← H*I ;                                                  303

  SUM3X1 ← SUM3X1 + (XNU[I]×HTI) ;                             304

  SUM3Y1 ← SUM3Y1 + (YNU[I]×HTI) ;                             305

END ;                                                          306

X1 ← XT1 + SUM3X1 ;                                            307

Y1 ← YT1 + SUM3Y1 ;                                            308

SUM4DX1 ← 0 ;                                                  309

SUM4DY1 ← 0 ;                                                  310

FOR I ← 1 STEP 1 UNTIL MP2 DO                                  311

BEGIN                                                          312
```

```
            IHTI ← I×(H*(I - 1)) ;                                                    313
            SUM4DX1 ← SUM4DX1 + (XNU[I]×IHTI) ;                                       314
            SUM4DY1 ← SUM4DY1 + (YNU[I]×IHTI) ;                                       315
          END ;                                                                       316
          DX1 ← DXT1 + SUM4DX1 ;                                                      317
          DY1 ← DYT1 + SUM4DY1 ;                                                      318
          T ← T + H ;                                                                 319
          Y1SQ ← Y1*2 ;                                                               320
          R ← SQRT((X1+MU)*2 + Y1SQ) ;                                                321
          S ← SQRT((X1-MUPM)*2 + Y1SQ) ;                                              322
          J ← (0.5)×(DX1*2 + DY1*2 - X1*2 - Y1*2) -MUPM/R - MU/S ;                    323
          DJ ← ABS(J - J1) ;                                                          324
          J1 ← J ;                                                                    325
TEST:     IF T < TTEST THEN GO TO SWAP ;                                              326
          DUB ← FALSE ;                                                               327
          IF SIGN(Y1) = SAME THEN GO TO SWAP ;                                        328
          IF ABS(Y1) > EPS1 THEN                                                      329
          BEGIN                                                                       330
            T ← T - H ;                                                               331
            GO TO HALVE                                                               332
          END ;                                                                       333
          IF NOT ITER THEN GO TO QUIT ;                                               334
          IF ABS(DX1) < EPS2 THEN                                                     335
          BEGIN                                                                       336
            ITER ← FALSE ;                                                            337
            TTEST ← (0.9)×PER ;                                                       338
            SAME ● SIGN(DIR2) ;                                                       339
            WRITE(PRINT[PAGE]) ;                                                      340
            CC ← 1 ;                                                                  341
            GO TO ITERATE                                                             342
          END ;                                                                       343
INTERP:DYHOLD ● ((DYA×DX1)-(DXA×DYSTART)) / (DX1-DXA) ;                               344
          DXA ● DX1 ;                                                                 345
```

```
        DYA ← DYSTART ;                                                          346
        DYSTART ← DYHOLD ;                                                       347
        GO TO ITERATE ;                                                          348
SWAP:   X0 ← X1 ;                                                                349
        Y0 ← Y1 ;                                                                350
        DX0 ← DX1 ;                                                              351
        DY0 ← DY1 ;                                                              352
        IF ITER THEN GO TO STEST ;                                               353
        IF CC =11 THEN                                                           354
        BEGIN                                                                    355
          CC ← 1 ;                                                               356
          WRITE(PRINT[PAGE1)                                                     357
        END ;                                                                    358
        WRITE(PRINT,LFMT,PLINE1) ;                                               359
        IF PLT THEN                                                              360
        BEGIN                                                                    361
          PLOTX[NP] ← X0 ;                                                       362
          PLOTY[NP] ← Y0 ;                                                       363
          NP ← NP + 1                                                            364
        END ;                                                                    365
STEST:  IF (R<RMIN) OR (R>RMAX) OR (S<SMIN) OR(S>SMAX) OR (T>TMAX) THEN          366
        GO TO START ;                                                            367
        IF NOT DUB THEN GO TO TSTEP ;                                            368
        IF (ERRX < EPSM) AND (ERRY < EPSM) THEN H ← 2×H ;                        369
        GO TO TSTEP ;                                                            370
QUIT:   IF CC = 11 THEN WRITE(PRINT[PAGE]) ;                                     371
        WRITE(PRINT,LFMT,PLINE) ;                                                372
        IF PLT THEN                                                              373
        BEGIN                                                                    374
          PLOTX[NP] ← X1 ;                                                       375
          PLOTY[NP] ← Y1 ;                                                       376
          IDENT(FRAMEID,DATE) ;                                                  377
          QUIK3V(-1,26,ABSCISSA,ORDINATE,NP,PLOTX,PLOTY) ;                       378
          ENDJOB ;                                                               379
```

```
          END :                                                          380
          GO TO START :                                                  381
  EOF:    CLOSE(PCARD,RELEASE) :                                         382
 END.                                                                    383
9110T0300120000000000PCARD0010000001  05PCARD                                9
931                                                                           9
 LABEL  0000000003 0PCARD0010000001
+0.1212856276 5@-01
+0.30000000@+01+0.16574301@-01+0.30000000@+01+0.45212967@-02+0.70000000@+01
+0.00000000@+00
+0.10000000@-06+0.10000000@-08+0.10000000@-08
+0.12500000@+00
+04
+0.62 000000@+01-1+1
+0.12000000000@+01+0.000000000000@+00+0.000000 0000@+00-0.10490000000@+01
 TRUE TRUE TRUE
06216@
TEST    DR. E. FEHLBERG C.P. REED
931                                                                           9
```

IMPROVED INITIAL CONDITIONS

The following is a listing of successive sets of initial conditions as obtained through the optional iterative interpolation procedure and based on example input values.

```
 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04900000000D+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01

 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04895672410+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01

 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04928511920+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01

 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04935706010+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01

 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04934543640+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01

 T =  0.00000000000D+00  X =  1.20000000000D+00  Y =  0.00000000000D+00 XD =  0.00000000000D+00 YD = -1.04934543400+00
DT =  1.25000000000D-01  J =  0.00000000000D+00 DJ =  0.00000000000D+00 RE =  1.21212856280+00 RM =  2.12128562780-01
```

EXAMPLE ORBIT

Based on the improved initial conditions, the following listing describes a complete orbit. The notation is as follows:

T is the time,

X and Y are the position coordinates,

XD and YD are velocity components,

DT is the step size,

J is the Jacobi constant,

DJ is the difference between successive Jacobi constants,

RE is the distance to the Earth's center, and

RM is the distance to the Moon's center.

T = 0.00000000000@+00 X = 1.20000000000@+00 Y = 0.00000000000@+00 XD = 0.00000000000@+00 YD = -1.04934543400@+00
DT = 1.25000000000@-01 J = 0.00000000000@+00 DJ = 0.00000000000@+00 RE = 1.21212856260@+00 RM = 2.12126562780@-01

T = 1.25000000000@-01 X = 1.18582307970@+00 Y = -1.29740060680@-01 XD = -2.23859022870@-01 YD = -1.01580578270@+00
DT = 1.25000000000@-01 J = -1.04160169440@+00 DJ = 1.04160169440@+00 RE = 1.20495668850@+00 RM = 2.36679817680@-01

T = 2.50000000000@-01 X = 1.14513226960@+00 Y = -2.51742817250@-01 XD = -4.22392311660@-01 YD = -9.29004185440@-01
DT = 1.25000000000@-01 J = -1.04160157660@+00 DJ = 1.17870513350@-07 RE = 1.18432558030@+00 RM = 2.96825563980@-01

T = 3.75000000000@-01 X = 1.08139001290@+00 Y = -3.60394342020@-01 XD = -5.92939017800@-01 YD = -8.03518501040@-01
DT = 1.25000000000@-01 J = -1.04160158590@+00 DJ = 9.32777766140@-09 RE = 1.15137611450@+00 RM = 3.72330237490@-01

T = 6.25000000000@-01 X = 8.98670591590@-01 Y = -5.20002343410@-01 XD = -8.48926398910@-01 YD = -4.51599244070@-01
DT = 2.50000000000@-01 J = -1.04160160230@+00 DJ = 1.63854565470@-08 RE = 1.04878860440@+00 RM = 5.27597600470@-01

T = 8.75000000000@-01 X = 6.67890670140@-01 Y = -5.75477232730@-01 XD = -9.73567011170@-01 YD = 2.98700307140@-02
DT = 2.50000000000@-01 J = -1.04160156260@+00 DJ = 3.96394170820@-08 RE = 8.90842411720@-01 RM = 6.58454050560@-01

T = 1.12500000000@+00 X = 4.23857445350@-01 Y = -4.91712198450@-01 XD = -9.57598885270@-01 YD = 6.78324266230@-01
DT = 2.50000000000@-01 J = -1.04160129370@+00 DJ = 2.68904841510@-07 RE = 6.57164123620@-01 RM = 7.48259760470@-01

T = 1.25000000000@+00 X = 3.06255414690@-01 Y = -3.79542725310@-01 XD = -9.28656150940@-01 YD = 1.14521191260@+00
DT = 1.25000000000@-01 J = -1.04160143220@+00 DJ = 1.38519681060@-07 RE = 4.95399877580@-01 RM = 7.80162213770@-01

T = 1.31250000000@+00 X = 2.47838351000@-01 Y = -2.98089790140@-01 XD = -9.50317639330@-01 YD = 1.47973515130@+00
DT = 6.25000000000@-02 J = -1.04160145150@+00 DJ = 1.92958395930@-08 RE = 3.95525371200@-01 RM = 7.97813569530@-01

T = 1.37500000000@+00 X = 1.84903259660@-01 Y = -1.90893313630@-01 XD = -1.10587523130@+00 YD = 1.99892162010@+00
DT = 6.25000000000@-02 J = -1.04160323470@+00 DJ = 1.78316258830@-06 RE = 2.74338834730@-01 RM = 8.25347291380@-01

T = 1.40625000000@+00 X = 1.46739034710@-01 Y = -1.22135334380@-01 XD = -1.38306476080@+00 YD = 2.43483595000@+00
DT = 3.12500000000@-02 J = -1.04160363380@+00 DJ = 3.99115378970@-07 RE = 2.00389504290@-01 RM = 8.49953386070@-01

T = 1.42187500000@+00 X = 1.22924055270@-01 Y = -8.17137841770@-02 XD = -1.50052610130@+00 YD = 2.75524298530@+00
DT = 1.56250000000@-02 J = -1.04160365770@+00 DJ = 2.38505890590@-08 RE = 1.57849143690@-01 RM = 8.65798662580@-01

T = 1.43750000000@+00 X = 9.16889455730@-02 Y = -3.54224894770@-02 XD = -2.40525249360@+00 YD = 3.19055185670@+00
DT = 1.56250000000@-02 J = -1.04160484580@+00 DJ = 1.20817276190@-06 RE = 1.09694244980@-01 RM = 8.96882272720@-01

T = 1.44531250000@+00 X = 7.02093922400@-02 Y = -9.56328913940@-03 XD = -3.17631498220@+00 YD = 3.41973171430@+00
DT = 7.81250000000@-03 J = -1.04160488060@+00 DJ = 1.47410901260@-08 RE = 8.28914671940@-02 RM = 9.17711874900@-01

T = 1.44921875000@+00 X = 5.66202059250@-02 Y = 3.92064419660@-03 XD = -3.82355031230@+00 YD = 3.46677025870@+00
DT = 3.90625000000@-03 J = -1.04160487880@+00 DJ = 1.71712599690@-09 RE = 6.88604723140@-02 RM = 9.31259484390@-01

T = 1.45312500000@+00 X = 3.99125987310@-02 Y = 1.72660994450@-02 XD = -4.29875434220@+00 YD = 3.30904337530@+00
DT = 3.90625000000@-03 J = -1.04160488960@+00 DJ = 3.92319634560@-08 RE = 5.48306545640@-02 RM = 9.48116067620@-01

T = 1.45703125000@+00 X = 1.85379901230@-02 Y = 2.89482314320@-02 XD = -5.21034291130@+00 YD = 2.49495288700@+00

DT = 3.90625000000E-03 J = -1.04160488572E+00 DJ = 1.75350578508-08 RE = 4.21715255728-02 RM = 9.69765606638-01

T = 1.45898437500E+00 X = 5.66720293098-03 Y = 3.29734606618-02 XD = -6.94781018008+00 YD = 1.54930939258+00
DT = 1.95312500000E-03 J = -1.04160486158+00 DJ = 4.30736690768-09 RE = 3.74691657868-02 RM = 9.82757552508-01

T = 1.46093750000E+00 X = -8.40709856718-03 Y = 3.47036348578-02 XD = -7.38403364218+00 YD = 1.60529915528-01
DT = 1.95312500000E-03 J = -1.04160482478+00 DJ = 3.67581378678-08 RE = 3.49026011658-02 RM = 9.96842773048-01

T = 1.46289062508+00 X = -2.27811694008-02 Y = 3.34922384958-02 XD = -7.22746521568+00 YD = -1.38644174708+00
DT = 1.95312500000E-03 J = -1.04160488598+00 DJ = 3.47063178208-08 RE = 3.51455269928-02 RM = 1.01120740778+00

T = 1.46484375008+00 X = -3.62901313928-02 Y = 2.94925001398-02 XD = -6.54517108678+00 YD = -2.63150986228+00
DT = 1.95312500000E-03 J = -1.04160488577+00 DJ = 1.74622982748-09 RE = 3.81259617988-02 RM = 1.02458612448+00

T = 1.46679687508+00 X = -4.82331449248-02 Y = 2.35341376818-02 XD = -5.68076472378+00 YD = -3.39345328188+00
DT = 1.95312500000E-03 J = -1.04160488552+00 DJ = 2.47382558888-09 RE = 4.30975230068-02 RM = 1.03637182568+00

T = 1.46875000008+00 X = -5.85256916348-02 Y = 1.44840659448-02 XD = -4.87945334238+00 YD = -3.77633095758+00
DT = 1.95312500000E-03 J = -1.04160488552+00 DJ = 0.00000000000E+00 RE = 4.92383793128-02 RM = 1.04652695898+00

T = 1.47265625008+00 X = -7.50833875106E-02 Y = 1.20228255128-03 XD = -3.68712373618+00 YD = -3.96584239748+00
DT = 3.90625000000E-03 J = -1.04160490068+00 DJ = 4.53874235978-08 RE = 6.29667915288-02 RM = 1.06295599238+00

T = 1.47656250008+00 X = -8.78983996448-02 Y = -1.41526139748-02 XD = -2.92727432998+00 YD = -3.87452191328+00
DT = 3.90625000000E-03 J = -1.04160491508+00 DJ = 1.44354999078-08 RE = 7.70763162098-02 RM = 1.07585892778+00

T = 1.48046875008+00 X = -9.82852911548-02 Y = -2.89844543918-02 XD = -2.42513625828+00 YD = -3.71523180588+00
DT = 3.90625000000E-03 J = -1.04160491408+00 DJ = 9.74978320318-10 RE = 9.09014877578-02 RM = 1.08654338958+00

T = 1.48828125008+00 X = -1.14623838318-01 Y = -5.67216818368-02 XD = -1.82117897498+00 YD = -3.39181548168+00
DT = 7.81250000000E-03 J = -1.04160492488+00 DJ = 8.77480488278-09 RE = 1.17143632788-01 RM = 1.10395340288+00

T = 1.49609375008+00 X = -1.27407542058-01 Y = -8.21199477528-02 XD = -1.48016293078+00 YD = -3.11924659258+00
DT = 7.81250000000E-03 J = -1.04160493188+00 DJ = 6.67932908988-09 RE = 1.41537741158-01 RM = 1.11829821198+00

T = 1.50390625008+00 X = -1.39074225688-01 Y = -1.05584547448-01 XD = -1.26614612578+00 YD = -2.89597905838+00
DT = 7.81250000000E-03 J = -1.04160493178+00 DJ = 8.14907252798-10 RE = 1.64352597218-01 RM = 1.13088770048+00

T = 1.51953125008+00 X = -1.55723725778-01 Y = -1.48015633858-01 XD = -1.02133653598+00 YD = -2.55371089348+00
DT = 1.56250000008-02 J = -1.04160475758+00 DJ = 1.59823688958-07 RE = 2.06223661848-01 RM = 1.15313421808+00

T = 1.53515625008+00 X = -1.70584697778-01 Y = -1.85849963778-01 XD = -8.93842275098-01 YD = -2.30038514898+00
DT = 1.56250000008-02 J = -1.04160475158+00 DJ = 5.92262984978-09 RE = 2.44231835738-01 RM = 1.17327126438+00

T = 1.55468625008+00 X = -1.94444852018-01 Y = -2.51716310338-01 XD = -7.81661424888-01 YD = -1.93923466758+00
DT = 3.12500000008-02 J = -1.04160405408+00 DJ = 6.95523340268-07 RE = 3.11983325468-01 RM = 1.21077090078+00

T = 1.59765625008+00 X = -2.20230075308-01 Y = -3.08125210588-01 XD = -7.48335894008-01 YD = -1.68328367858+00
DT = 3.12500000008-02 J = -1.04160404258+00 DJ = 1.35623848938-08 RE = 3.71816332248-01 RM = 1.24677600548+00

T = 1.6289062500D+00 X = -2.4354005879D-01 Y = -3.5751098401D-01 XD = -7.4693577326D-01 YD = -1.4846650863D+00
DT = 3.1250000000D-02 J = -1.0416040418D+00 DJ = 6.5483618528D-10 RE = 4.2587014940D-01 RM = 1.2822590909D+00

T = 1.6914062500D+00 X = -2.9111758821D-01 Y = -4.4032849815D-01 XD = -7.8012897120D-01 YD = -1.1811009341D+00
DT = 6.2500000000D-02 J = -1.0416039599D+00 DJ = 8.1941834651D-08 RE = 5.2127158239D-01 RM = 1.3526648194D+00

T = 1.7539062500D+00 X = -3.4131338594D-01 Y = -5.0653744712D-01 XD = -8.2662520074D-01 YD = -9.4558984370D-01
DT = 6.2500000000D-02 J = -1.0416039586D+00 DJ = 1.2514647096D-09 RE = 4.0410498521D-01 RM = 1.4224318893D+00

T = 1.8789062500D+00 X = -4.5009537918D-01 Y = -6.0042402610D-01 XD = -9.0925024515D-01 YD = -5.7201205564D-01
DT = 1.2500000000D-01 J = -1.0416040790D+00 DJ = 1.2035889085D-07 RE = 7.4318499945D-01 RM = 1.5582867439D+00

T = 2.0039062500D+00 X = -5.6713232547D-01 Y = -6.5221601175D-01 XD = -9.5672562150D-01 YD = -2.6442448772D-01
DT = 1.2500000000D-01 J = -1.0416040879D+00 DJ = 8.9203240350D-09 RE = 8.5639646344D-01 RM = 1.6862450676D+00

T = 2.2539062500D+00 X = -8.0644576429D-01 Y = -6.5222016974D-01 XD = -9.3154937016D-01 YD = 2.4453444898D-01
DT = 2.5000000000D-01 J = -1.0416055378D+00 DJ = 1.4498400738D-06 RE = 1.0277796293D+00 RM = 1.9091792397D+00

T = 2.5039062500D+00 X = -1.0208192920D+00 Y = -5.3936745973D-01 XD = -7.6186014574D-01 YD = 6.3898113536D-01
DT = 2.5000000000D-01 J = -1.0416055493D+00 DJ = 3.1504896469D-08 RE = 1.1438418789D+00 RM = 2.0798451150D+00

T = 2.7539062500D+00 X = -1.1781670171D+00 Y = -3.4306273669D-01 XD = -4.8195369389D-01 YD = 9.0931476527D-01
DT = 2.5000000000D-01 J = -1.0416055707D+00 DJ = 1.4697434381D-09 RE = 1.2154578225D+00 RM = 2.1930377623D+00

T = 3.2539062500D+00 X = -1.2442138499D+00 Y = 1.6405171289D-01 XD = 2.3003339541D-01 YD = 1.0194315464D+00
DT = 5.0000000000D-01 J = -1.0416056574D+00 DJ = 3.8417056203D-09 RE = 1.2429590174D+00 RM = 2.2381058270D+00

T = 3.7539062500D+00 X = -9.4690398033D-01 Y = 5.7832716094D-01 XD = 8.1867743545D-01 YD = 5.4673904933D-01
DT = 5.0000000000D-01 J = -1.0416056607D+00 DJ = 1.1388328858D-07 RE = 1.1179809953D+00 RM = 2.0404490560D+00

T = 4.0039062500D+00 X = -7.4453220238D-01 Y = 6.6029749139D-01 XD = 9.5325285500D-01 YD = 1.2219932439D-01
DT = 2.5000000000D-01 J = -1.0416056553D+00 DJ = 5.4278643802D-09 RE = 9.8879042979D-01 RM = 1.8554012486D+00

T = 4.2539062500D+00 X = -5.0492998813D-01 Y = 6.2981173447D-01 XD = 9.3674444484D-01 YD = -4.2011963048D-01
DT = 2.5000000000D-01 J = -1.0416056537D+00 DJ = 8.2945916802D-08 RE = 7.9949748388D-01 RM = 1.6202218726D+00

T = 4.3789062500D+00 X = -3.9147457879D-01 Y = 5.5493266756D-01 XD = 8.6932752122D-01 YD = -7.5661587941D-01
DT = 1.2500000000D-01 J = -1.0416056537D+00 DJ = 1.0748417269D-09 RE = 6.7396509680D-01 RM = 1.4877234232D+00

T = 4.5039062500D+00 X = -2.8667388150D-01 Y = 4.3660129327D-01 XD = 7.7941537988D-01 YD = -1.1943400248D+00
DT = 1.2500000000D-01 J = -1.0416056541D+00 DJ = 7.0358510129D-08 RE = 5.1681524997D-01 RM = 1.3491436692D+00

T = 4.5664062500D+00 X = -2.4119827000D-01 Y = 3.5282372475D-01 XD = 7.4623464114D-01 YD = -1.5027600141D+00
DT = 6.2500000000D-02 J = -1.0416056544D+00 DJ = 7.9890014603D-09 RE = 4.2066318551D-01 RM = 1.2787090944D+00

T = 4.6289062500D+00 X = -1.9398127243D-01 Y = 2.4559377466D-01 XD = 7.8808735725D-01 YD = -1.9695627469D+00
DT = 6.2500000000D-02 J = -1.0416056973D+00 DJ = 5.2341743885D-07 RE = 3.0558545952D-01 RM = 1.2071004939D+00

T = 4.6601562500D+00 X = -1.4775223092D-01 Y = 1.7856052201D-01 XD = 9.1374312184D-01 YD = -2.3460671640D+00

```
DT =  3.12500000000E-02  J = -1.04160602970E+00 DJ =  5.66506059840E-08 RE =  2.36859844870E-01 RM =  1.16933738610E+00

 T =  4.69140625000E+00  X = -1.33989265870E-01  Y =  9.63707308390E-02 XD =  1.34141892820E+00 YD = -2.98046690870E+00
DT =  3.12500000000E-02  J = -1.04161200660E+00 DJ =  5.97498090130E-06 RE =  1.55361992540E-01 RM =  1.12599234230E+00

 T =  4.70703125000E+00  X = -1.08611575540E-01  Y =  4.58851525920E-02 XD =  2.01932776460E+00 YD = -3.51680672100E+00
DT =  1.56250000000E-02  J = -1.04161357630E+00 DJ =  1.56964233610E-06 RE =  1.06838284270E-01 RM =  1.09744268400E+00

 T =  4.71484375000E+00  X = -9.01001524850E-02  Y =  1.71193615120E-02 XD =  2.81233886880E+00 YD = -3.84553979600E+00
DT =  7.81250000000E-03  J = -1.04161358370E+00 DJ =  7.43602868170E-09 RE =  7.98288252580E-02 RM =  1.07810751820E+00

 T =  4.71875000000E+00  X = -7.78493136940E-02  Y =  1.84383999290E-03 XD =  3.50985626970E+00 YD = -3.95946677950E+00
DT =  3.90625000000E-03  J = -1.04161358000E+00 DJ =  3.75439412890E-09 RE =  6.57466109290E-02 RM =  1.06572234600E+00

 T =  4.72265625000E+00  X = -6.21685186760E-02  Y = -1.35483499610E-02 XD =  4.60140008930E+00 YD = -3.85774038870E+00
DT =  3.90625000000E-03  J = -1.04161357070E+00 DJ =  7.29633029550E-08 RE =  5.18416336000E-02 RM =  1.05012735740E+00

 T =  4.72656250000E+00  X = -4.11934522190E-02  Y = -2.73285157270E-02 XD =  6.20852764870E+00 YD = -2.98671445060E+00
DT =  3.90625000000E-03  J = -1.04161459400E+00 DJ =  1.08701351570E-06 RE =  3.98950569720E-02 RM =  1.02942770240E+00

 T =  4.72851562500E+00  X = -2.82563304000E-02  Y = -3.22132512010E-02 XD =  7.00392470030E+00 YD = -1.93163466900E+00
DT =  1.95312500000E-03  J = -1.04161459880E+00 DJ =  5.80621417610E-09 RE =  3.60249697890E-02 RM =  1.01643825140E+00

 T =  4.73046875000E+00  X = -1.40944695510E-02  Y = -3.45910210410E-02 XD =  7.39992629340E+00 YD = -4.56576401020E-01
DT =  1.95312500000E-03  J = -1.04161455070E+00 DJ =  3.74711817130E-08 RE =  3.46469534180E-02 RM =  1.00256482360E+00

 T =  4.73242187500E+00  X =  2.34521576440E-04  Y = -3.39769193080E-02 XD =  7.17550965590E+00 YD =  1.04835906710E+00
DT =  1.95312500000E-03  J = -1.04161460610E+00 DJ =  5.53700374440E-08 RE =  3.61562843790E-02 RM =  9.88221183850E-01

 T =  4.73437500000E+00  X =  1.36459934230E-02  Y = -3.07493914280E-02 XD =  6.51469533280E+00 YD =  2.17800853250E+00
DT =  1.95312500000E-03  J = -1.04161459100E+00 DJ =  1.50757841770E-08 RE =  4.01229713530E-02 RM =  9.74710592070E-01

 T =  4.73632812500E+00  X =  2.56149503720E-02  Y = -2.57635755290E-02 XD =  5.74362779210E+00 YD =  2.86155973930E+00
DT =  1.95312500000E-03  J = -1.04161459180E+00 DJ =  8.00355337560E-10 RE =  4.56982998370E-02 RM =  9.62611323670E-01

 T =  4.73828125000E+00  X =  4.53903570650E-02  Y = -1.32919534350E-02 XD =  4.45883990330E+00 YD =  3.39572416800E+00
DT =  1.95312500000E-03  J = -1.04161471470E+00 DJ =  1.22876372190E-07 RE =  5.90347538510E-02 RM =  9.42574804750E-01

 T =  4.74023437500E+00  X =  6.10132644010E-02  Y =  1.84227962480E-04 XD =  3.59970052780E+00 YD =  3.44427841480E+00
DT =  1.95312500000E-03  J = -1.04161473970E+00 DJ =  2.49856388470E-08 RE =  7.31420591810E-02 RM =  9.26858191140E-01

 T =  4.74218687500E+00  X =  7.38793822110E-02  Y =  1.35954906040E-02 XD =  3.02393218800E+00 YD =  3.39038202260E+00
DT =  1.95312500000E-03  J = -1.04161474210E+00 DJ =  2.41561792790E-09 RE =  8.70758517840E-02 RM =  9.14093164840E-01

 T =  4.74609375000E+00  X =  9.48594923130E-02  Y =  2.66210279420E-02 XD =  2.62021619330E+00 YD =  3.27572941450E+00
DT =  3.90625000000E-03  J = -1.04161474220E+00 DJ =  4.36557456850E-11 RE =  1.00575348630E-01 RM =  9.03404057220E-01

 T =  4.75097656250E+00  X =  1.03111017290E-01  Y =  5.12640852910E-02 XD =  2.09963982120E+00 YD =  3.03538322150E+00
DT =  7.81250000000E-03  J = -1.04161474220E+00 DJ =  2.16823536900E-08 RE =  1.26127583230E-01 RM =  8.86244327010E-01
```

T = 4.76757812500E+00 X = 1.18186051000E-01 Y = 7.41231403970E-02 XD = 1.78295948400E+00 YD = 2.42200077190E+00
DT = 7.81250000000E-03 J = -1.04161471760E+00 DJ = 2.88127921520E-09 RE = 1.49920440580E-01 RM = 8.72838422030E-01

T = 4.78320312500E+00 X = 1.42919325720E-01 Y = 1.15440346250E-01 XD = 1.42459220650E+00 YD = 2.44403530270E+00
DT = 1.56250000000E-02 J = -1.04161415700E+00 DJ = 5.60583430340E-07 RE = 1.93303702160E-01 RM = 8.52801585540E-01

T = 4.79882812500E+00 X = 1.63549915800E-01 Y = 1.52184252170E-01 XD = 1.23387042180E+00 YD = 2.23016084930E+00
DT = 1.56250000000E-02 J = -1.04161414140E+00 DJ = 1.55705492940E-08 RE = 2.32428428640E-01 RM = 8.38251762490E-01

T = 4.83007812500E+00 X = 1.98784591610E-01 Y = 2.15800402230E-01 XD = 1.04939871830E+00 YD = 1.86515946830E+00
DT = 3.12500000000E-02 J = -1.04161264570E+00 DJ = 1.49571860680E-06 RE = 3.01753236850E-01 RM = 8.18061554650E-01

T = 4.86132812500E+00 X = 2.30215722680E-01 Y = 2.69821025920E-01 XD = 9.72712939380E-01 YD = 1.60444457170E+00
DT = 3.12500000000E-02 J = -1.04161262210E+00 DJ = 2.35886545850E-08 RE = 3.62676355330E-01 RM = 8.04267099810E-01

T = 4.89257812500E+00 X = 2.60032987210E-01 Y = 3.16665836350E-01 XD = 9.40480549890E-01 YD = 1.40087893750E+00
DT = 3.12500000000E-02 J = -1.04161262110E+00 DJ = 1.04773789640E-09 RE = 4.17551387490E-01 RM = 7.93741810190E-01

T = 4.95507812500E+00 X = 3.18242789540E-01 Y = 3.93943999000E-01 XD = 9.29559252800E-01 YD = 1.08775871770E+00
DT = 6.25000000000E-02 J = -1.04161250270E+00 DJ = 1.18365278870E-07 RE = 5.14137243130E-01 RM = 7.76913380090E-01

T = 5.01757812500E+00 X = 3.76715959780E-01 Y = 4.54046504510E-01 XD = 9.43334822080E-01 YD = 8.43730584610E-01
DT = 6.25000000000E-02 J = -1.04161250100E+00 DJ = 1.73167791220E-09 RE = 5.97794523360E-01 RM = 7.41340128900E-01

T = 5.14257812500E+00 X = 4.96776845690E-01 Y = 5.34328850150E-01 XD = 9.75222014120E-01 YD = 4.57129942210E-01
DT = 1.25000000000E-01 J = -1.04161263190E+00 DJ = 1.30845373650E-07 RE = 7.37897035400E-01 RM = 7.25728040600E-01

T = 5.26757812500E+00 X = 6.19393575360E-01 Y = 5.71236566710E-01 XD = 9.81028001330E-01 YD = 1.42072790290E-01
DT = 1.25000000000E-01 J = -1.04161264180E+00 DJ = 9.92440618570E-09 RE = 8.51586557940E-01 RM = 6.79770008230E-01

T = 5.39257812500E+00 X = 7.40534518910E-01 Y = 5.71634530120E-01 XD = 9.51090416870E-01 YD = -1.29430988050E-01
DT = 1.25000000000E-01 J = -1.04161264270E+00 DJ = 9.16770659300E-10 RE = 9.45131454830E-01 RM = 6.22854159140E-01

T = 5.44257812500E+00 X = 9.60318174510E-01 Y = 4.81357431310E-01 XD = 7.84436109810E-01 YD = -5.71819016470E-01
DT = 2.50000000000E-01 J = -1.04161284290E+00 DJ = 2.00244905460E-07 RE = 1.08506111970E+00 RM = 4.82145371200E-01

T = 5.49257812500E+00 X = 1.12239579400E+00 Y = 2.96712046650E-01 XD = 4.93308803130E-01 YD = -8.83456524470E-01
DT = 2.50000000000E-01 J = -1.04161295250E+00 DJ = 1.09575921670E-07 RE = 1.17268220530E+00 RM = 3.25743426810E-01

T = 5.51757812500E+00 X = 1.17245022350E+00 Y = 1.79423477100E-01 XD = 3.06070855970E-01 YD = -9.86601180330E-01
DT = 1.25000000000E-01 J = -1.04161294890E+00 DJ = 3.65253072230E-09 RE = 1.19828850900E+00 RM = 2.57561212420E-01

T = 5.54257812500E+00 X = 1.19773723140E+00 Y = 5.19540443440E-02 XD = 9.08349165830E-02 YD = -1.04379658450E+00
DT = 1.25000000000E-01 J = -1.04161297030E+00 DJ = 2.14054647301E-08 RE = 1.21097079550E+00 RM = 2.14101302240E-01

T = 5.57382812500E+00 X = 1.19947531830E+00 Y = 1.92496556490E-02 XD = 3.37267506580E-02 YD = -1.04856725870E+00
DT = 3.12500000000E-02 J = -1.04161297030E+00 DJ = 2.32830643660E-10 RE = 1.21195674290E+00 RM = 2.12676828100E-01

T = 5.58945312500E+00 X = 1.19997786000E+00 Y = 2.85851659420E-03 XD = 4.98928826590E-03 YD = -1.04931408000E+00

DT = 1.56250000000E-02 J = -1.04161297030E+00 DJ = 1.45519152280E-11 RE = 1.21210979340E+00 RM = 2.12125683720E-01

T = 6.19140625000E+00 X = 1.19998409410E+00 Y = 8.09057145150E-04 XD = 1.39438473840E-03 YD = -1.04932950480E+00
DT = 1.95312500000E-03 J = -1.04161297030E+00 DJ = 2.91038304570E-11 RE = 1.21211292690E+00 RM = 2.12114199830E-01

T = 6.19189453130E+00 X = 1.19998455550E+00 Y = 2.96688899050E-04 XD = 4.95642461850E-04 YD = -1.04933065050E+00
DT = 4.88281250000E-04 J = -1.04161297030E+00 DJ = 1.45519152280E-11 RE = 1.21211315460E+00 RM = 2.12113325770E-01

T = 6.19213867190E+00 X = 1.19998462170E+00 Y = 4.05046323880E-05 XD = 4.62706112270E-05 YD = -1.04933081660E+00
DT = 2.44140625000E-04 J = -1.04161297030E+00 DJ = 1.45519152280E-11 RE = 1.21211318510E+00 RM = 2.12113188300E-01

T = 6.19214919950E+00 X = 1.19998462220E+00 Y = 8.49159716950E-06 XD = -9.90088021770E-06 YD = -1.04933081850E+00
DT = 3.05175781250E-05 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318500E+00 RM = 2.12113185150E-01

T = 6.19217681890E+00 X = 1.19998462210E+00 Y = 4.75838361770E-07 XD = -2.39437531310E-05 YD = -1.04933081830E+00
DT = 7.62939453130E-06 J = -1.04161297030E+00 DJ = 1.45519152280E-11 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217705730E+00 X = 1.19998462210E+00 Y = 2.25654399040E-07 XD = -2.43825929100E-05 YD = -1.04933081830E+00
DT = 2.38418579100E-07 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217717650E+00 X = 1.19998462210E+00 Y = 1.00568417710E-07 XD = -2.46020127990E-05 YD = -1.04933081830E+00
DT = 1.19209289550E-07 J = -1.04161297030E+00 DJ = 1.45519152280E-11 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217723610E+00 X = 1.19998462210E+00 Y = 3.80234270340E-08 XD = -2.47117227440E-05 YD = -1.04933081830E+00
DT = 5.96046447750E-08 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217726590E+00 X = 1.19998462210E+00 Y = 6.75093169560E-09 XD = -2.47665777160E-05 YD = -1.04933081830E+00
DT = 2.98023223880E-08 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217728079E+00 X = 1.19998462210E+00 Y = 2.84186977830E-09 XD = -2.47734345480E-05 YD = -1.04933081830E+00
DT = 1.49011611940E-08 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217728715E+00 X = 1.19998462210E+00 Y = 8.87338819720E-10 XD = -2.47768630240E-05 YD = -1.04933081830E+00
DT = 1.86264514920E-09 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

T = 6.19217728743E+00 X = 1.19998462210E+00 Y = -8.99244595090E-11 XD = -2.47785772410E-05 YD = -1.04933081830E+00
DT = 9.31322574620E-10 J = -1.04161297030E+00 DJ = 0.00000000000E+00 RE = 1.21211318490E+00 RM = 2.12113184850E-01

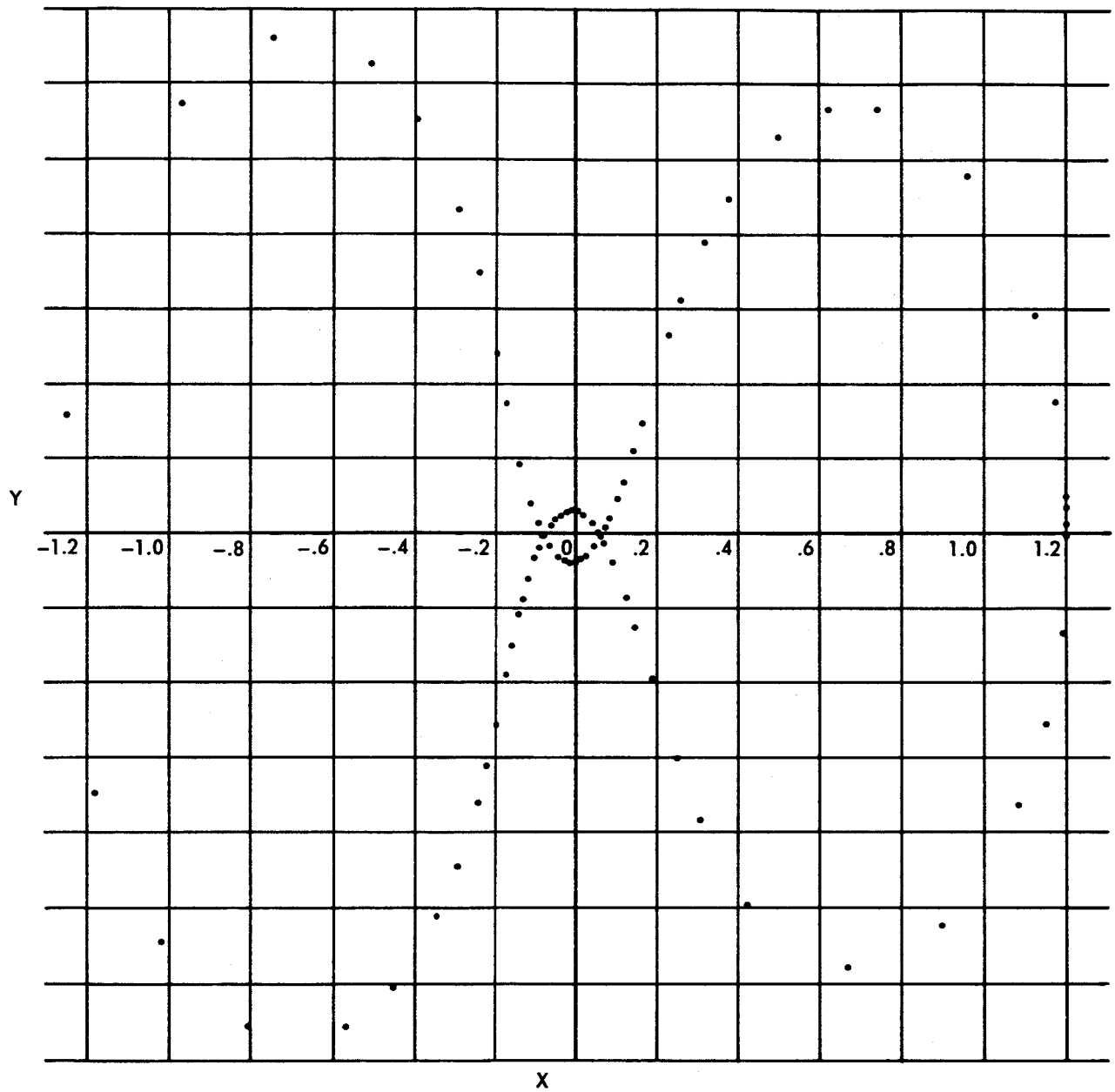


Figure 2. SC 4020 Plot of Final Orbit.