EUR 3622 e

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

SAHYB-2: A PROGRAMME FOR THE SOLUTION OF DIFFERENTIAL EQUATIONS USING AN ANALOGUE-ORIENTED LANGUAGE

by

H. d'HOOP and R. MONTEROSSO

1967

Joint Nuclear Research Center
Ispra Establishment - Italy
Scientific Information Processing Center - CETIS

This report is on sale at the addresses listed on cover page 4

at the price of FF 12.50	FB 125.–	DM 10.–	Lit. 1.560	Fl. 9.–

When ordering, please quote the EUR number and the title, which are indicated on the cover of each report.

Printed by SMEETS
Brussels, September 1967

EUR 3622 e

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

SAHYB-2: A PROGRAMME FOR THE SOLUTION OF DIFFERENTIAL EQUATIONS USING AN ANALOGUE-ORIENTED LANGUAGE

by

H. d'HOOP and R. MONTEROSSO

1967

Joint Nuclear Research Center
Ispra Establishment - Italy
Scientific Information Processing Center - CETIS

SUMMARY

Many programmes and sub-programmes exist for the integration of differential equations. Nevertheless for the study of practical cases (as encountered in nuclear reactors simulations) their use is subject to many restrictions, or needs an important programming effort.

SAHYB-2 is derived from the experience of a previous programme SAHYB-1, and is designed to be used by physicists or engineers who are not specialists in programming. In particular it is addressed to analogue computation engineers who need an easy way to obtain digital solutions to be compared with analog results.

The problem description is concentrated in one sub-routine for initial value problems and in two sub-routines for boundary value problems.

Other sub-routines are already prepared to handle operations such as :

— Output of results
— Modification of integration and/or print-out step
— Interpolation of tabulated functions
— Fixed or variable delays
— Plot of curves on CALCOMP plotter.

The existing versions are for the IBM 7090 and the IBM 360/65 computers.

KEYWORDS

DIFFERENTIAL EQUATIONS
PROGRAMMING
INTEGRALS
MATHEMATICS
S-CODES
DIGITAL SYSTEMS
ANALOG SYSTEMS
COMPUTERS
NUMERICALS
IBM 7090
IBM 360

CONTENTS

SAHYB-2: a programme for the solution of differential equations using an analogue-oriented language (+)

1) Introduction

The programme SAHYB-2 has developed the idea contained in the programme SAHYB-1*, but is a new programme for the IBM-360/65 computer. **

SAHYB-2 uses as language a sub-set of FORTRAN-360 statements, and a certain number of functions intended to describe in a simple way scientific problems which require integration.

The problem description is brought as close to the mathematical formulae as permitted by the limitations of FORTRAN and is a set of ordinary differential equations reduced to the first order, and a set of algebraic relations; the system may be integrated for initial conditions, or may be solved for boundary conditions by a trial-and-error strategy. SAHYB-2 is intended to help the computer user in writing his problem in a simple and straight forward way, providing also a set of functions and subroutines corresponding to the most frequent requirements for this class of problems. Input and Output operations are treated in a simplified way, releasing the user from the need of complicate programming; in particular SAHYB-2 may be visualised as the simulation of the analogue and hybrid computer on the digital, since the user may think his programme in terms of parallel relations.

* Euratom report:
EUR 2519.e
SAHYB (Simulation of Analogue and Hybrid computer)
A GENERAL PURPOSE PROGRAMME FOR THE SOLUTION OF INITIAL AND BOUNDARY VALUE PROBLEMS

and

I.E.E.E. Transactions on Electric Computers;
June 1966: SAHYB, a programme for simulation of Analog and Hybrid computers,

by d'Hoop and Monterosso

** A version of SAHYB-2 for the IBM 7090 also exists.

(+)Manuscript received on July 7, 1967.

The integration method may be chosen between constant step 4^{th} order Runge-Kutta, or fixed or variable step Adams-Moulton predictor corrector. *

A certain number of diagnostics are included in the programme: these indicate a wrong arrangement of input data or an error in using subprogrammes.

2) General organisation

Preliminary remark: for the sake of clarity, the independent variable will always be referred to as TIME or T; the variables which must be integrated will be called X_i or X(I), and their derivatives DX_i or DX(I). Initial conditions are called XZ_i or XZ(I), and integration always begins at time zero.

The general organisation of SAHYB-2 is given by fig. 1:)

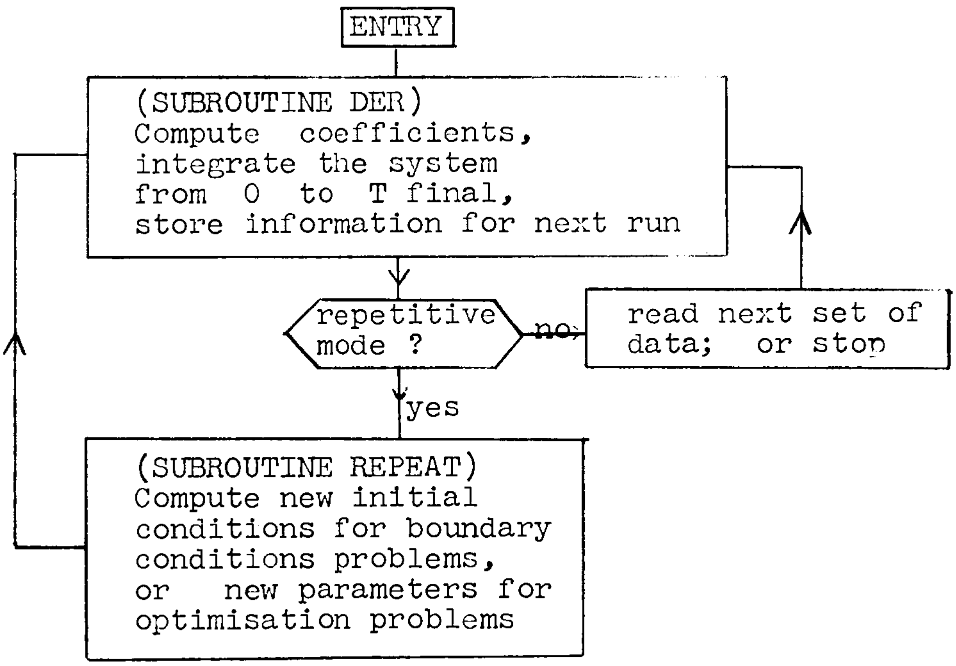

figure 1.

* The integration routines have been written by Mrs. P. Wood-Sangermano (Euratom, Ispra). For the mathematical method see:

- F.B. Hildebrand: Introduction to numerical analysis (chapter 6). McGraw-Hill 1956.
- R. Cansey and W.L. Frank: Share routine D2-RW-INT.

The problem must be described in two user-written subroutines, named DER and REPEAT, corresponding to the two main blocks of figure 1. DER contains the equations, and REPEAT the trial-and-error strategy to modify initial conditions (in order to match boundary conditions) or to modify the parameters to be optimised.

A certain number of subprogrammes are available to the user for specific tasks.

3) The subroutine DER

This subroutine must be supplied by the user. DER contains the equations, the coefficient description, and may call several subroutines and functions which will be described later; it should be written as follows:

```
  SUBRØUTINE  DER (T, X, DX, PAR)
  DIMENSIØN  X(n), DX(n), PAR(m)  etc...
  LØGICAL SWITCH
  IF (SWITCH( i )) GØ  TØ  100

    first group of statements

100  CØNTINUE

    second group of statements

  RETURN
  END
```

where: T, X, DX are as described in section 2; PAR is a vector of parameters, which is read in input (see sec.9.) and may be modified by the REPEAT loop (see figure 1).

Throughout this report the following graphical convention is used:

Ø is the letter O

0 is a zero

At the beginning of computation, for T=0, the first and the second group of statements are executed sequentially; the first group of statements contains the preliminary computations of coefficients, which need to be computed only once; the test:

```
IF (SWITCH ( i )) GO TO 100
```

by-passes the first group of statements during the following iterations (see sec 6); i is an integer from 1 to 5.

The second group of statements contains the differential equations, written as:

```
DX(I) = expression ,
```

which may be mixed with explicit algebraic relation between variables.

The use of DER will be easily understood considering the following simple example:

$$\ddot{x} = - \dot{x} - (2\pi fr)^2 x$$

$$z = x^2 + \dot{x}^2$$

$$\dot{y} = z + 5$$

the first group of statements should contain:

```
DELTA = 0.001
PI    = 3.1416926
FR    = 0.2
A     = (2. * PI * FR)** 2
```

The second group of statements is executed at each iteration of the integration. It should contain the differential equations *, and other algebraic or logical relations associated with the variables of the problem.

* i.e. the expression of the derivatives.

In the same example:

```
DX(1) = - DELTA * X(1) - A * X(2)
DX(2) = X(1)
Z     = X(2) ** 2 + X(1) ** 2
DX(3) = Z + 5
```

Note that all the integrated variables have a common name X(I), being the components of a vector. The same is true for the derivatives DX(I).

Other statements, to be explained later, can be included in this second group of instructions.

The maximum allowed number of differential equations is 150.

This set of statements can be visualised as an analogue block diagram, if desired; then each DX(I) can be thought of as an integrator, where the inputs appear on the right hand side.

It should be noted that the statements are executed in sequence and that an auxiliary equation, such as Z = ..., must be written before an equation where Z appear in the right hand side *.

The initial conditions are given as a set of input data (see section 9).

4) The subroutine REPEAT

This routine must be supplied by the user.

* Since this programme is executed without rearranging the equations, the auxiliary quantity needs to be defined before its use. This is obvious to a digital programmer used to FORTRAN, but is not the case in other systems such as APACHE or MIDAS, used by analogue engineers.

When boundary value or optimisation problems are computed using the trial and error technique, computation is repeated several times, changing some initial conditions or some parameters as functions of final values of the computation.

The law for changing the initial conditions should be contained in another subroutine called REPEAT, which should have the following structure:

```
      SUBROUTINE REPEAT (X, XZ, PAR)
      DIMENSION X(n), XZ(n)
      LOGICAL SWITCH
      IF(SWITCH(m)) GØ TØ 100

      first group of statements

  100 CONTINUE

      second group of statements

      RETURN
      END
```

where: X, XZ and PAR have the meaning described in sections 2 and 3; m is an integer from 5 to 10 (see section 6).

The first group of statements is executed only once, at the first iteration on the initial conditions.

It may be used for instance to initialize a count of iterations.

The second group of statements is executed at each iteration, i.e. at the end of each integration run. It contains the formulae expressing the new initial conditions as function of the final values (X) and the initial values (XZ) of the preceding run.

For example:

```
X(4) = XZ(4) - 0.008 * X(2)
PAR(5) = A + X(2) + PAR(6)
```

All the initial conditions which are not changed should be re-defined by a statement such as:

```
X(I) = XZ(I)
```

When the repeat loop is not used (initial value problems), the routine should be written as follows:

```
SUBRØUTINE REPEAT (.X, XZ, PAR)
CALL STØPR
RETURN
END
```

5) Output of results

Five subroutines are available to read out results in a standard arrangement, without any need of WRITE or FORMAT instructions. They must be called in the first group of statements of DER, which are executed only once. (This call does not produce output at the time in which the CALL statement is executed, but serves to inizialise the output procedures).

All results are printed at fixed time intervals given in the input data. *

* For the variable step mode, results will be printed out at the end of the first iteration after the desired time. Usually the time step for output is much bigger than the time step for integration.

These subroutines are: OUTPUT, WRITE, CØEF, DRAW, DRAWXY.
These routines have a variable number of arguments:

5.1 - CALL ØUTPUT (arg 1, arg 2, ...)

This subroutine enables to print out either the complete integrated vector and the derivatives, or some of them. Specified labels (given in the input data) are printed out near the result.

The arguments are integers, and are the indexes of components which must be printed out.

For example:

CALL ØUTPUT (2, 4, 7)

prints out only X(2), X(4), X(7) and the corresponding derivatives.

The maximum number of arguments is 150.

<u>If OUTPUT is not called the complete vector is printed out.</u>

5.2 - CALL WRITE (arg_{11}, arg_{12},... arg_{11}, arg_{12},...)

This subroutine prints out variables other than the integrated vector components. These variables must appear at least once in a left hand side term in the subroutine DER. They are printed out for the same interval as in OUTPUT. The number of arguments is variable with a minimum of two; they are always in pairs, the first one specifying the label, the second one the variable name.

arg_{i1} is a Hollerith field of a maximum of 4 characters, which specify the label that should be printed near the variable.

arg_{i2} is a name of a variable which must be printed out.

For example:

CALL WRITE (4HDELT , DELT , 1HZ, Z)

The maximum number of pairs of arguments is 50 .

5.3 - CALL CØEF (arg_{11}, arg_{12},... arg_{i1}, arg_{i2})

This routine prints out numerical values of coefficients that have been computed in the first group of statements of DER; the arguments arrangement is the same as in WRITE (section 5.2). The maximum number of pairs of arguments is 50 .

5.4 - CALL DRAW (arg_{11}, arg_{12}, arg_{i1}, arg_{i2},...)

This subroutine plots the results on a CALCOMP 506 digital plotter as a function of the independent variable T. The arguments have the same arrangement as for WRITE.

<u>Example:</u> CALL DRAW (4HX(1), X(1), 4HDELT , DELT)

The curves are plotted following a fixed format on linear scales and with zero at the origin.

For economy no indications are given on the axis, but the full scale values are printed out on the output listing for each curve.

Each curve is traced with its own appropriate scale; the full scale values are always one-digit numbers multiplied by a power of ten. For example a curve with a maximum value of 0,63 will have a full scale of $7 * 10^{-1}$.

The same for the independent variable (time).
Ex.: if the final time is T = 750 the full scale will be $8 * 10^{2}$.

Curves are numbered according to the order in which they appear as arguments of DRAW.

The number of curves is limited to 15, and the number of points per curve to 127 (only the first 127 points will be traced).

The size of the drawing is 40 x 40 cm. A comment is added to the drawing.

<u>If REPEAT is used, the output will be one drawing per iteration, so the use of DRAW should be prohibited for problems where convergence is still doubtful.</u>

The subroutine DRAW may be called repeatedly, with different arguments. For each CALL, the curves relative to the arguments will be drawn on the same plot. Thus

```
    CALL DRAW (1HA, A, 1HB, B)
```

will draw A and B on the same drawing,
and

```
    CALL DRAW (1HA, A)
    CALL DRAW (1HB, B)
```

will draw A and B on two separate drawings.

The total number of pairs of arguments for all the DRAW statements is 15. Thus a maximum number of 15 curves may be plotted, either on the same or on separate plots.

5.5 - CALL DRAWXY (arg_{11}, arg_{12},... arg_{11}, arg_{12})

This subroutine plots the results on the CALCOMP 506 digital plotter in a phase plane, taking as abscissa the first variable (first two arguments) and as ordinate the successive. The routine may be called repeatedly to obtain different plots in a similar way as DRAW.

For example:

```
CALL DRAWXY (1HA, A, 1HB, B, 1HC, C)
CALL DRAWXY (1HD, D, 1HE, E)
```

will trace two plots, the first giving B and C as function of A, and the second E as function of D.

The maximum number of pairs of arguments (thus of variables to be plotted) is 6. The maximum number of points per curve 127. The size of the drawing is 40 x 40 cm. *

6) Subprogrammes available for control purpose

These routines are built in the programme, and may be called in DER or in REPEAT, for various control operations.

6.1 - Logical function SWITCH (i)
where i is an integer constant, from 1 to 10. This function, associated with a logical IF statement, separates the instructions to be executed only once

* important remark: The routines DRAW and DRAWXY may be used only if the basic routines are available in the system for the CALCOMP; see for that report EUR 2280F, P. Moinil, J. Pire: Programmation relative on CALCOMP, which describes these routines. The routines needed for SAHYB 2 are: FINIM, PLOT, SYMBL4, LINE, FINTRA.

from those to be executed repeatedly.

SAHYB initialise SWITCH as .FALSE. at the first call; SWITCH becomes .TRUE. when called the first time and remains in this state. SWITCH may be used with arguments from 1 to 5 in DER, and is reset each time T = 0. SWITCH with argument from 6 to 10 may be used in repeat, and is reset only once. When the REPEAT loop is not used, the arguments from 1 to 10 may be used in DER. The main use of SWITCH has already been illustrated in sections 3 and 4, but many other uses may be found. The argument i must never be greater than 10.

6.2 - CALL MØDDT(X)
where X is a real floating point constant or variable. When the integration routine is used with constant step length, and MØDDT is called, the time step is modified and its new value is set to X. If the variable step option is utilised, MØDDT does not affect computation.

6.3 - CALL MØDDX(Y)
where Y is a real floating point constant or variable. The output of results proceed with a certain time step between prints. When MØDDX is called, the reat-out step is modified and its new value is set to Y. This affects all the routines mentioned in section 5 (except obviously PARAM).

6.4 - CALL START(Z)
where Z is a real floating point constant or variable. This routine may be used only in DER. When the predictor-corrector integration option is used, in some cases (as near a discontinuity), the integration method loses accuracy (or significance), unless the starting formula (RUNGE-KUTTA) is used again.*
CALL START (Z) starts again the predictor-corrector at the time Z.

* See Hildebrand: Introduction to numerical analysis

6.5 - CALL STØP(Z)

where Z is a real floating point constant or variable. This routine may be used only in DER. When called, it stops integration at the time Z, but the REPEAT loop (if any) continues; this routine should be used instead of the usual STØP of Fortran, also in order to save the results to be printed or plotted successively.

6.6 - CALL STØPR

This routine has no argument, and may be used only in REPEAT. It interrupts the REPEAT loop (fig. 1) and must be called by the REPEAT subroutine when the required conditions have been matched. Here again, this routine must be used instead of the usual Fortran STØP in order to save results to be printed or plotted successively.

When the REPEAT loop is not wanted, as for initial value problems, CALL STØPR is the only instruction of subroutine REPEAT (see section 4).

7) <u>Subprogrammes available for arithmetical or logical operations</u>

These routines are built in the programme, and may be called by DER or REPEAT; they are, in general, commonly used operations, and are given to release the user from the need to reprogramme them for every problem.

7.1 - SIGMA(X, I1, I2) and SIGMAP(X, Y, I1, I2)

where X and Y are dimensioned vectors of real floating point variables, and I1 and I2 are integers (I1 < I2). These functions perform a multiple summation, for example

A = SIGMA(X, 2, 6) + SIGMAP(X, Y, 1, 10)

is equivalent to

$$A = \sum_{i=2}^{6} X_i + \sum_{i=1}^{10} X_i * Y_i$$

7.2 - STEP(Z)

where Z is a real floating point constant or variable. This function may be used only in DER, and has a value zero for $T < Z$, and one for $T > Z$; when the transition from one state to another occurs, the current integration step is terminated to Z exactly, and, if used, the predictor-corrector is started again from this point, avoiding errors due to the discontinuity.

7.3 - PULSE(Z)

where Z is a real floating point constant or variable. This function may be used only in DER, and is equivalent to a pulse of unit surface occurring at the time Z. Before the pulse (which has the duration of one integration step) occurs, the current integration step is terminated to Z exactly, and, if used, the predictor corrector is started again from this point, avoiding errors due to the discontinuity.

7.4 - RAMP(Z1, Z2)

where Z1 and Z2 are real floating point constants or variables.

This function may be used only in DER, and is equivalent to a ramp of one unit per second, starting at time Z1 and ending at time Z2.

Thus

$$RAMP = 0. \quad \text{for } T < Z1$$

$$RAMP = T-Z_1 \quad \text{for } Z2 > T > Z1$$

$$RAMP = Z_2-Z_1 \quad \text{for } T > Z2$$

At times Z1 and Z2 the current integration steps are terminated exactly to Z1 or Z2, and, if used, the predictor-corrector is started again from these points, avoiding thus any error due to the discontinuity.

The interval between Z1 and Z2 should not be shorter than three integration steps.

7.5 - CØMPAR(Z, I)

where Z is a real floating point variable, and I is an integer constant from 1 to 10.
This function may be used only in DER, and acts as a comparator in an analogue computer:

CØMPAR(Z, I) = 0. for $Z < 0$

CØMPAR(Z, I) = 1. for $Z > 0$,

I being the comparator number, since several of them may be used simultaneously (up to 10). At each transition from one state to another the predictor-corrector, if used, is started again in order to avoid errors due to the discontinuity.

When Z changes sign, the transition of COMPAR occurs at the end of the current integration interval, without modifying the integration step length.

8) Subprogrammes available to read out or load numerical data in tables

These routines are built in the programme, and may be used by DER only. All the subprogrammes described in this section utilize a common storage, organised in 30 tables of 100 points each* (thus 100 abscissae and 100 ordinates). They are used for function generation, delay or memory; when used simultaneously, care should be taken to avoid erroneous overlap of data.

8.1 - Tabulated functions**

In engineering problems some functions may be obtained experimentally, and thus be given as sets of points.

*10 tables only for the 7090 version

**These would be on the diode function generators on an analogue computer

A maximum of 30 arbitrary functions, having a maximum of 100 points each, may be included in the input data (see section 9). The points are expressed as a set of coordinates x, y. These tables which have an identification number from 1 to 30, are selected by two functions to be used in the subroutine DER.

These functions are:

- for linear interpolation : TABLE (K, W)
- for parabolic interpolation : PAINT (K, W)

K is the integer ($\leqslant 30$) that identifies the table in which the main programme has stored the function to extract (see "input data" sec. 9).

W is the abscissa on which interpolation must be performed and can be any FORTRAN floating expression.

For example, these statements could be written in DER:

```
DX (2)  =  TABLE (2, X(1))
XX = PAINT (5, X(2) + X(4)) + DELTA
```

If one of the variables falls outside a given table, computation is stopped and a diagnostic is printed out.

8.2 -Functions of two variables

When a function of two variables is given under tabulated form:

$$y = f(x, z) \qquad (8\text{-}2\text{-}1)$$

it can be stored as a set of intersections:

$$y_1 = f_1 (x, z_1) \qquad (8\text{-}2\text{-}2)$$
$$\cdots\cdots\cdots\cdots$$
$$y_n = f_n (x, z_n)$$

each of which can be stored as a function of a single variable. In SAHYB2 these can be stored as tables.

If care is taken to store functions of successive (increasing) z_i in tables of successive indices K (see section 9), the function of two variables can be read by the function:

FTWØV(I1,I2,X,Z,VZ)

where:

I1 and I2 are integers, identifying the first and the last table in which the functions $f_i(x,z_i)$ are stored.

X and Z are real floating point variables, and have the meaning of x and z of (8-2-1).

VZ is a dimensioned vector, and corresponds to the z_i of (8-2-2).

for example, the following equations must be programmed:

$$A = B + C$$

$$B = f(x,z)$$

suppose that B is stored as three functions of one variable, for

$$z_1 = 0,5 \qquad z_2 = 1,5 \qquad z_3 = 3,6$$

These functions have been loaded in tables 3, 4, 5. The statements necessary to describe this operation in DER will be:

```
DIMENSION VZ(3)
VZ(1) = 0.5
VZ(2) = 1.5
VZ(3) = 3.6
    A = FTWØV(3,5,X,Z,VZ) + C
.............
```

Linear interpolation is used in both directions; a warning is given and computation is stopped if either X or Z fall out of the tabulated range.

8.3 - Fixed or variable delay lines

In many engineering problems delayed functions are necessary; for this purpose a function named DELAY is available in the programme, and is called as follows:

```
DELAY (K, TAU, STEP, XX)
```

where:

TAU is the time delay, and may be any FORTRAN floating expression, it can have a constant or a variable value indifferently.

STEP is the time interval at which the delayed function is stored. Its value must remain constant, and the ratio TAU/STEP should not be greater than 99 for storage limitations.

XX is the actual value of the variable to be delayed.

K is an integer ($\leqslant 30$) which specifies the identification number of the table used to store the delayed function.

To avoid overlap with the tabulated functions in input, K is subject to the following restriction: It must be greater (and never equal) than the identification number of the tables in input (used in TABLE and PAINT). (A good practice is to use high values of K, i.e. 30, 29, etc.... .)

Example:

If, in a dynamic system, a variable XX needs to be delayed by a time TAU, DELAY is to be used (obviously called in the DER subroutine). This function loads a table with a specified time step (STEP), and extracts it after a time delay TAU, using linear interpolation:

DX(2) = DELAY(10, TAU, STEP, XX) - A(2)

where TAU is the time delay and may vary and be any reasonable expression; STEP is related to the frequency domain pertinent to the problem.

When the last point of the table is reached, loading continues at the lower end, so that the table may be used several times.

If the table capacity (100 points) is exceeded (due to the ratio STEP/TAU), a diagnostic is printed out and computation is stopped.

8.4 - Loading tables during computation

When a function of time need to be memorised during computation, in order to be utilised later (i.e. after one turn around the REPEAT loop of fig. 1), this can be done calling:

CALL LØAD(K, STEP, X)

where:

K — is an integer (≤ 30) that identifies the table.

STEP — is a floating-point constant, and is the time interval for which the information must be stored.

X — is a floating-point variable, to be recorded in the table.

The table K may or not have been loaded (initialised) in input; to read the table the functions TABLE or PAINT may be used.

.5 - Least-Squares curve-fitting and polynomial generation

Sometimes a function is available in tabulated form, but its use may be more economical or more accurate using a polynomial approximation instead of table look-up and interpolation.

If the table has been loaded in input (as for TABLE, PAINT etc...) and the coefficients of a polynomial of order n are wanted, these can be found calling:

```
CALL FIT(K, N, A)
```

(this subroutine must be called only once, in the first group of statements of SUBROUTINE DER) where:

K is an integer (≤ 30) that identifies the table.

N is the order of the polynomial wanted (integer).

A is a dimensioned floating-point vector (dimension N+1), and represents the coefficients of the polynomial.

for example:

Suppose a least-square best fit of order 3 is wanted to generate a function f(x), which is tabulated in table 5. The DER will be written as follows:

```
     LØGICAL SWITCH
     DIMENSIØN A(4)
     IF(SWITCH(1)) GØ TØ 100
     ..........
     CALL FIT (5, 3, A)
     ..........
100  CØNTINUE
     ..........
     F= ((A(4) * X + A(3)) * X + A(2)) * X + A(1)
     ..........
```

The subroutine FIT may also be called in REPEAT.

9) Input data arrangement

The first card is a comment intended to identify the problem. the first 15 letters of this comment are written on the CALCOMP plot for identification. On the following, there is one numerical information per card: columns 1 to 40 may be used for any comment to be reproduced on the output listing (usually the meaning of that particular datum). The numerical information is contained in columns 41 to 50, in fixed or floating-point notation.*

The data cards are, in order:

1) identification and comments
2) number of differential equations N (fixed point maximum 150)
3) number of parameters to be varied in REPEAT loop (fixed point, maximum 100)
4) number of tables to be read in input (fixed point, maximum 30)
5) final time for integration (floating)
6) initial or constant integration step (floating)
7) time step for output (floating)

* For fixed point the reading format is : I 10
for floating point the reading format is : E 10.6
Thus for fixed point the unities should be in column 50; for floating if the decimal point is explicit, the number may be written in any desired way in columns 41 to 47; column 48 contains eventually the sign of the exponent of 10, columns 49 and 50 this exponent.
Blanks are taken as zeroes.

8) integration option (maximum 4, fixed point).
The numerical information in column 50 should be:

1 for fixed step predictor-corrector
2 for variable step predictor-corrector
3 for variable step predictor-corrector using standard error characteristics*
4 for 4th order Runge-Kutta integration

9,1) initial condition for X(1) in floating.
<u>The first six characters of the comment are taken as label on the standard output.</u>
....

9,n) initial condition for X(n)

10,1) first parameter to be optimised by repeat loop (initial value), in floating.
....

10,m) parameter m.

<u>Then, the following cards (11 to 16) must be placed only when integration option is 2 on datum n.8.</u>

11) maximum relative error (floating)

12) minimum time step (floating)

13) maximum time step (floating)

14) variation factor (floating)

15) level below which the absolute error is considered for step variation (floating)

16) error factor, specifying a lower bound for which a longer step length is taken (floating)

* These are: maximum error: 10^{-5}
minimum time step: 10^{-5}
maximum time step: 100
variation factor : 0.5
level below which the absolute error is considered: 1
error factor for which a longer step length is taken: 100

EURATOM - CETIS

DATA (Zones de 10 colonnes)

PROBLEM	DATE	PAGE OF

Columns 1–80

Line	Columns 1–40	Columns 41–50	Remarks
1	IDENTIFICATION HEADING		
2	NUMBER OF DIFFERENTIAL EQUATIONS	XXX	
3	NUMBER OF PARAMETERS TO BE VARIED	XXX	
4	NUMBER OF TABLES IN INPUT	XX	
5	FINAL TIME	X.XXXXX±XX	1 fixed step predictor-correct.
6	(INITIAL) INTEGRATION STEP	X.XXXXX±XX	2 variable step pred.-corr.
7	STEP FOR OUTPUT AND PLOTS	X.XXXXX±XX	3 v. st. pred-corr stand data
8	INTEGRATION OPTION NUMBER	X	← 4 4th order Runge-Kutta.
9	NAME 1 (INITIAL CONDITION OF X(1))	±X.XXXX±XX	
10	etc....		maximum 150 cards
11	PAR 1 (IF REPEAT IS USED)	±X.XXXX±XX	
12	etc....		maximum 100 cards
13	MAXIMUM ERROR	X.XXXXX±XX	only with option 2 (lines 13–18)
14	MINIMUM TIME STEP	X.XXXXX±XX	
15	MAXIMUM TIME STEP	X.XXXXX±XX	
16	VARIATION FACTOR	X.XXXXX±XX	
17	LIMIT BETWEEN ABS. AND REL. ERROR	X.XXXXX±XX	
18	LOWER ERROR BOUND	X.XXXXX±XX	
19	NUMBER OF POINTS IN TABLE 1	XX	
20	← x1 → ← y1 → ← x2 → ← y2 →	← x3 →	← y3 →
21	other tables if any		
22	data related to a read statement in DER		
23	second set of data, etc...		
24	END		
25	end of file		

CET - 002

figure 2

If the number of tables is not zero, the following cards must then be given:

17) number of points in table 1 ($\leqslant$ 100, fixed point)

18-1) the following cards contain the abscissae and ordinates of table one: six numbers, per card; respectively x_1, y_1, x_2, y_2, x_3, y_3 on the first card, etc. Each number is expressed in floating and uses 10 columns. Thus columns 1 to 60 are used on these cards.

19) number of points in table 2 (fixed point)

19-1) abscissae and ordinates of table two.

And so on for the desired number of tables.

If a second case must be computed, a second deck of data must follow in the same order; if not, the last card must always contain END in the first three columns.

A résumé of the data arrangement is given in figure 2.

10) Miscellaneous information

SAHYB 2 is made by a main programme and 27 subprogrammes, not including DER and REPEAT, to be supplied by the user. Comments are given throughout the programme to facilitate understanding.

10-1) Unused subprogrammes

The subprogrammes DELAY, TABLE, PAINT, FIT, FTWØV, LØAD, RAMP, STEP, PULSE, SIGMA, SIGMAP, CØMPAR, when not used by DER or REPEAT, may be removed to economise loading time and storage (FTWØV needs TABLE).

To save further storage, if none of the subprogrammes described in section 8 are used, the COMMON labeled/ELBAT/ in the main programme may be removed.

10-2) Calcomp plotter

If a CALCOMP digital plotter is not available, the routine CURVE must be replaced by a dummy subroutine:

```
SUBRØUTINE CURVE
RETURN
END
```

and SCALA and VALMAX become then unnecessary. If a plotter different than the CALCOMP is available, it should be easy to modify subroutine CURVE following the comments: other parts of the programme would not need to be changed.

10-3) Integration subroutines

These subroutines are:

NRM1 for Runge-Kutta integration

NADM for the Adams-Moulton predictor-corrector (NADM needs NRM1 to start).

These routines have been selected for SAHYB 2 as representing integration methods widely tested and suited to a great variety of scientific problems. Nevertheless, the use of other routines supplied by the user remains possible and easy to insert, following these indications:

1) The new subroutine must communicate with the programme by means of a common:
COMMON/VARIAB/NEQ,T,X(150),DX(150), DT,E,F,HMIN, HMAX,BETA,A,IND

where:

NEQ is the number of differential equation
T is the independent variable
X is the integrated vector
DX is the vector of the derivatives
DT is the integration step
E,F,HMIN,HMAX,BETA,A are the data relative to variable step integration (see section 9)
IND is an indicator to be used freely.

2) In addition insert the common statement:

```
COMMON/MAP/PAR(100)
```

and the new subroutine must evaluate the derivatives by:

```
CALL DER (T,X,DX,PAR)
```

where the arguments are as explained in section 2 and 3 (note that PAR is not used by the integration routine, but is necessary as a means of communication between DER, REPEAT, and the main programme).

3) In the main programme, after the comment INTEGRATION ROUTINE SELECTION DEPENDING ON IOP the computed GO TO must be changed, and a new option number must be added; the new integration routine must be called immediately before the comment TEST FOR FINAL TIME OR START TIME.

for example:

```
    GØ TØ (351, 351, 351, 354, 9999), IØP
    ..........
9999 CALL INTEGR
    ..........
```

INTEGR will be used if the option number is 5 in data card n°. 8.

11) <u>Example</u>

These equations represent the distribution of Temperature and Velocity of a fluid moving between two parallel plane surfaces:

$$\frac{d}{dx} \quad (1 + \alpha V) * \lambda(T) * \frac{dT}{dx} = 0 \qquad (1)$$

$$\frac{d}{dx} \left[\mu(T) * \frac{dV}{dx}\right] = \Delta p + \gamma(T) \qquad (2)$$

$$\lambda = a - bT \qquad (3)$$

$$\gamma = c - dT \qquad (4)$$

$$\mu = e^{\frac{p}{T} + h} \qquad (5)$$

The boundary conditions are:

$T(0) = 100 \qquad T(x_{max}) = 0$

$V(0) = 0 \qquad V(x_{max}) = 0$

This problem must be solved by trial and error.

Let us define:

$X(1) = V \qquad DX(1) = \dot{V}$

$X(2) = T \qquad DX(2) = \dot{T}$

$X(3) = \left[\mu(T) * DX(1)\right] \qquad DX(3) = X\ (3)$

$PAR(1) = \left[(1 + \alpha V) * \lambda(T) * DX(2)\right]$

$\frac{1}{\mu(T)}$ = TABLE(1, X(2))

.and is tabulated. *

* This function is here tabulated to illustrate the use of TABLE.

From equations (1) to (5) we can derive the following system:

$$DX(1) = TABLE(1, X(2)) * X(3) \qquad (6)$$
$$DX(2) = X(4)/(1 + \alpha V) * \lambda \qquad (7)$$
$$DX(3) = \Delta p + \gamma \qquad (8)$$

let us call:

α = ALFA
γ = GA
W = $(1 + \alpha * X(1)) * \lambda$
λ = VA
Δp = DP

Equations (6) to (8) will easily be recognized in the subroutine DER that follows. Since initial conditions are given for X(1) and X(2), the SUBROUTINE REPEAT changes the initial conditions for X(3) and the value of PAR(1) until the boundary conditions are matched for X(1) and X(2).

Note that the final values of X(1) and X(2) are considered to match the boundary conditions (=0) if their final value is less than 1 % of their peak value. If the number of iterations (NIT) exceeds 5, computation is here stopped. PAR(2) and PAR(3) are variation factors which are not modified within the programme. They provide here an easy way for the user to act manually on the problem, changing data cards between computations, seeking for the best conditions for convergence.

Following the DER and REPEAT will be found the input data for this problem.

EURATOM - CETIS

DATA (Zones de 10 colonnes)

PROBLEM *data cards for Sahyb* DATE *1967* PAGE *1* OF *1*

```
JOE  SMITH          BOUNDARY VALUE PROBLEM
NUMBER OF DIFF. EQUATIONS                        3
NUMBER OF PARAMETERS                             3
N.  OF  TABLES                                   1
FINAL  TIME                             0.19
INTEGRATION   STEP                      1.9    -02
OUTPUT  STEP                            1.9    -02
INTEGRATION  OPTION (RUNGE-KUTTA)                4
VELOCITY    INITIAL CONDITION           0.0
TEMPERATURE      I. C.                  100.
X(3)             I. C.                  1.
PAR 1     TENTATIVE VALUE               -0.8
PAR 2     VARIATION FACT. FOR X(3)      10.
PAR 3     VARIATION FACT. FOR PAR(1)    6.     -03
N. OF POINTS IN TABLE 1                         14
-100.     0.        -20.      0.1       -10.      0.2
   0.     0.271     +11.      0.3295     20.      0.3724
  29.     0.4249     45.      0.5294     55.      0.6022
  67.     0.6974     78.      0.7922     90.      0.904
   1.  +02 1.004     2.    +02
END
```

CET - 002

```
      SUBROUTINE DER(T,X,DX,PAR)
C
C     EXAMPLE 1  BOUNDARY VALUE PROBLEM
C
      DIMENSION X(3),DX(3),PAR(3)
      LOGICAL SWITCH
C
      IF(SWITCH(1))  GO TO 100
C
      A=.15984E-2
      B=.8E-6
      C=1.29556
      D=.72E-3
      ALFA=1.
      DP=-2.
C
      CALL OUTPUT(1,2,3)
      CALL WRITE(2HVA,VA,2HGA,GA,1HW,W)
      CALL DRAW(2HX1,X(1),2HX2,X(2))
C
  100 CONTINUE
C
C     ***EQUATIONS***
      VA=A-B*X(2)
      GA=C-D*X(2)
      W=(1.+ALFA*X(1))*VA
      DX(1)=TABLE(1,X(2))*X(3)
      DX(2)=PAR(1)/W
      DX(3)=DP+GA
C
      RETURN
      END
```

```
      SUBROUTINE REPEAT(X,XZ,PAR)
C
C     X(1)  AND  X(2)  MUST BE NULL FOR T = T MAX
C
      DIMENSION X(3),XZ(3),PAR(3)
      LOGICAL SWITCH
C
      IF(SWITCH(6))  GO TO 100
C
      NIT=0
C
  100 CONTINUE
C
      NIT=NIT+1
      IF(NIT.GT.5) GO TO 130
C
C     ***TEST FOR BOUNDARY CONDITIONS***
      IF(ABS(X(1)).LT.0.00025) GO TO 110
C
C     ***MODIFY INITIAL CONDITION***
      X(3)=XZ(3)-X(1)*PAR(2)
      GO TO 120
  110 IF(ABS(X(2)).LT.2.) GO TO 130
C
C     ***MODIFY PARAMETER***
      PAR(1)=PAR(1)-X(2)*PAR(3)
      X(3)=XZ(3)
  120 X(1)=XZ(1)
      X(2)=XZ(2)
      RETURN
  130 CALL STOPR
      RETURN
      END
```


output of results:

```
TIME =  0.                              TIME STEP =  1.90000E-02

X(  1) = VELOCI =  0.                   DER(VELOCI) =  5.57444E-02    VA     =  1.51840E-03
X(  2) = TEMPER =  1.00000E 02          DER(TEMPER) = -5.36787E 02    GA     =  1.22356E 00
X(  3) = X(3)   =  5.55223E-02          DER(X(3)  ) = -7.76440E-01    W      =  1.51840E-03
------------------------------------------------------------------------------------------
TIME =  1.89999E-01                     TIME STEP =  1.90000E-02

X(  1) = VELOCI = -1.73130E-04          DER(VELOCI) = -2.34022E-02    VA     =  1.59782E-03
X(  2) = TEMPER =  7.19222E-01          DER(TEMPER) = -5.10192E 02    GA     =  1.29504E 00
X(  3) = X(3)   = -8.51529E-02          DER(X(3)  ) = -7.04958E-01    W      =  1.59755E-03
------------------------------------------------------------------------------------------

***CALCOMP DIGITAL PLOTTER   2 TIME CURVES,   0 X-Y CURVES, 10 POINTS PER CURVE***


************PLOT NUMBER  1          10   POINTS

            TIME , FULL SCALE =      2.00E-01

CURVE  1  X1     , FULL SCALE =      2.00E-03

CURVE  2  X2     , FULL SCALE =      1.00E 02
---------------------------------------------------

     ****  EXECUTION OF SAHYB COMPLETED  ****
```

Index (by sections)

Appendix I

Programme for the IBM 360/65.

The listing is given here to facilitate understanding of the text and should be taken as indicative, as in all programmes further corrections or improvements may be introduced.

```
C     S A H Y B 2 /   F O R T R A N 4  /  V E R S I O N   3 6 0 / 6 5
C     ****************************************************************
C
C
C                              HEAD  =  72 CHARACTERS  = HEADING
C                              CXZ  NAME OF INTEGRATED VARIABLES
C                              CXZ(10,I) HEADING OF VARIABLE X(I)=CHAR.1-6
C                              HDCURV(I)=NAME OF CURVE(I)
C
C     ****************************************************************
C     *    COMMONS   SUBPROGRAMS                                     *
C     *    CALCOM   /MAIN,STAMPA,PUTOUT,CURVE,VALMAX,SCALA           *
C     *    ELBAT    /MAIN,DELAY,TABLE,PAINT,FIT,LOAD                 *
C     *    GOLWS    /MAIN,COMPAR,SWLOG                               *
C     *    HCTIWS   /MAIN,SWITCH,MODDX,MODDT,FTWOV                   *
C     *    OLDVAL   /MAIN,COMPAR                                     *
C     *    RAP      /MAIN,NADM,NRM1                                  *
C     *    REP      /MAIN,STOPR                                      *
C     *    TRATS    /MAIN,COMPAR,MODDX,MODDT,START,STOP,PULSE        *
C     *    UTIL     /MAIN,STAMPA,PUTOUT                              *
C     *    VARIAB   /MAIN,NADM,NRM1,DELAY,COMPAR,START,STOP,PULSE,STAMPA*
C     *              RAMP,STEP,PUTOUT                                *
C     ****************************************************************
      COMMON/CALCOM/TABTX(128,16),HCURV(15),NCALL,NCURV(15),TABXY(128,6)
     1,HCURVX(6),NCALLX,NCURVX(6),KA,NA,NAXY,NOCALC,HEAD(18),NOXY
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT
      COMMON/GOLWS/SL(20)
      COMMON/HCTIWS/S(15)
      COMMON/OLDVAL/TOLD
      COMMON/RAP/PAR(100)
      COMMON/REP/NOREP
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD
      COMMON/UTIL/NS(150),N1,N2,N3,ID,VN(50),VE(50),CXZ(10,150)
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1
C
      DIMENSION XZ(150),XF(150)
      DIMENSION CNEQ(10),CTF(10),CDTZ(10),CDOUT(10),CIOP(10),
     1CNT(10),CNP(10),CNPAR(10),CPAR(10,100)
      DIMENSION CE(10),CHMIN(10),CHMAX(10),CBETA(10),CA(10),CF(10)
C
      LOGICAL NOCALC,NOREP,BEGIN,NOXY,SL,S,SWITCH
C
    1 FORMAT (18A4)
    2 FORMAT (10A4,I10)
    3 FORMAT (10A4,E10.6)
```

```
    4 FORMAT (6E10.6)                                                   MAIN0460
C                                                                       MAIN0470
   51 FORMAT (1H1,6X,18A4///)                                           MAIN0480
   52 FORMAT (1H0,6X,10A4,I5,14X,25HNUMBER OF DIFF. EQUATIONS/          MAIN0490
     11H0,6X,10A4, I5 , 14X ,32HN  OF PARAMETERS TO BE OPTIMISED /      MAIN0500
     21H0,6X,10A4,I5,14X,25HNUMBER OF TABLES IN INPUT /                 MAIN0510
     31H0,6X,10A4,1PE15.5,4X,10HFINAL TIME/                             MAIN0520
     41H0,6X,10A4,1PE15.5,4X,19H(INITIAL) TIME STEP/1H0,6X,10A4,1PE15.5,MAIN0530
     54X,20HTIME STEP FOR OUTPUT/1H0,6X,10A4,I5,14X,18HINTEGRATION OPTIOMAIN0540
     6N/)                                                               MAIN0550
   53 FORMAT (1H1,6X,42HPARAMETERS TO BE OPTIMISED BY REPEAT LOOP //(1H0MAIN0560
     1,6X, 10A4,3X,4HPAR(, I2,2H)=, 1PE12.5))                           MAIN0570
   54 FORMAT (1H0/1H0,27H** NUMBER OF TABLES EXCEEDS,I4,                MAIN0580
     116H NO EXECUTION **  /)                                           MAIN0590
   55 FORMAT (1H1,12X,10A4, 2X,28H** NUMBER OF POINTS IN TABLE,I4,5X,1H=MAIN0600
     1,I4,3H **/)                                                       MAIN0610
   56 FORMAT (1H0/1H0,42H** TABLE CAPACITY EXCEEDED NO EXECUTION ** )   MAIN0620
   57 FORMAT (1H1,6X,21HVARIABLE DEFINITIONS ,22X,18HINITIAL CONDITIONS,MAIN0630
     111X,19HINITIAL DERIVATIVES //                                     MAIN0640
     2(1H0, 6X,10A4, 3X,2HX(,I3,1H),1PE17.7,6X,3HDX(,I3,1H),1PE17.7))   MAIN0650
   58 FORMAT (1H1)                                                      MAIN0660
   59 FORMAT (1H1, 8X, 27H***CALCOMP DIGITAL PLOTTER  ,I3,14H TIME CURVEMAIN0670
     1S, ,I3,12H X-Y CURVES, I3,20H POINTS PER CURVE***  )              MAIN0680
   60 FORMAT (1H0//7X,24HREPEAT LOOP ITERATION No. ,I4,10H COMPLETED //) MAIN0690
   61 FORMAT (1H0, 6X,53H** NUMBER OF ITERATIONS EXCEEDS 25 EXECUTION ENMAIN0700
     1DED **/)                                                          MAIN0710
   62 FORMAT (1H0,1P2E25.7)                                             MAIN0720
   63 FORMAT(1H1, 6X,40H****  EXECUTION OF SAHYB COMPLETED  **** )      MAIN0730
   64 FORMAT (1H0///,7X,52HADDITIONAL DATA FOR VARIABLE STEP INTEGRATIONMAIN0740
     1 OPTION / 1H0/,7X,10A4,1PE15.5,4X,13HMAXIMUM ERROR/               MAIN0750
     21H0,6X,10A4,1PE15.5,4X,17HMINIMUM TIME STEP/                      MAIN0760
     31H0,6X,10A4,1PE15.5,4X,17HMAXIMUM TIME STEP/                      MAIN0770
     41H0,6X,10A4,1PE15.5,4X,16HVARIATION FACTOR/                       MAIN0780
     51H0,6X,10A4,1PE15.5,4X,33HLIMIT BETWEEN ABS. AND REL. ERROR/      MAIN0790
     61H0,6X,10A4,1PE15.5,4X,20HLOWER BOUND TO ERROR//)                 MAIN0800
   65 FORMAT (1H0///,7X,52HSTANDARD DATA FOR VARIABLE STEP INTEGRATION OMAIN0810
     1PTION   / 1H0/,7X,1PE15.5,4X,13HMAXIMUM ERROR/                    MAIN0820
     21H0,6X,1PE15.5,4X,17HMINIMUM TIME STEP/                           MAIN0830
     31H0,6X,1PE15.5,4X,17HMAXIMUM TIME STEP/                           MAIN0840
     41H0,6X,1PE15.5,4X,16HVARIATION FACTOR/                            MAIN0850
     51H0,6X,1PE15.5,4X,33HLIMIT BETWEEN ABS. AND REL  ERROR/           MAIN0860
     61H0,6X,1PE15.5,4X,20HLOWER BOUND TO ERROR//)                      MAIN0870
   66 FORMAT(1H0,6X,32HERROR IN INPUT DATA NO EXECUTION//)              MAIN0880
   67 FORMAT(1H0,6X,43HPREDICTOR-CORRECTOR STARTED AGAIN AT TIME =   ,  MAIN0890
```

```
     11PE12.5)
   68 FORMAT(1H0///,7X,33HCONSTANT STEP PREDICTOR-CORRECTOR)
   69 FORMAT(1H0///,7X,32HVARIBLE STEP PREDICTOR-CORRECTOR)
   70 FORMAT(1H0///,7X,34HCONSTANT STEP 4TH ORD. RUNGE-KUTTA)
C
C
C                        INITIALISATION ENTRY
C     ***************************************************************
      DATA END/3HEND/
      DATA MM,MMM/30,100/
C     MM IS THE NUMBER OF TABLES AVAILABLE
C     MMM IS THS MAXIMUM NUMBER OF POINTS PER TABLE
C     MM  AND  MMM  MUST BE CHANGED IF THE DIMENSIONS OF THE COMMON
C     /ELBAT/ ARE CHANGED
C
C                                    ************************
C                                    *                      *
C                                    * RETURN FOR NEXT DATA *
C                                    *                      *
C                                    ************************
  100 CONTINUE
      READ (5,1) HEAD
      IF(END.EQ.HEAD(1)) GO TO 101
      GO TO 103
  101 IF(NOCALC.OR.NOXY) GO TO 102
      CALL FINTRA
C     NOCALC AND NOXY ARE TRUE WHEN CALCOMP IS NOT USED
C     FINTRA ENDS THE USE OF CALCOMP
  102 WRITE(6,63)
      STOP
C
C     READ DATA
C
  103 CONTINUE
      READ (5,2) CNEQ,NEQ,CNPAR,NPAR ,CNT,NT
      READ (5,3) CTF,TE,CDTZ,DTZ,CDOUT,DOUTZ
      READ (5,2) CIOP,IOP
      IOP1=IOP-1
       IENTRY=0
      READ (5,3) ((CXZ(K,I),K=1,10),XZ(I),I=1,NEQ)
      IF(NPAR.EQ.0) GO TO 105
      READ (5,3) ((CPAR(K,I),K=1,10),PAR(I),I=1,NPAR)
C
C     WRITE DATA FOR CHECK
C
```

```
  105 WRITE (6,51) HEAD                                                   MAIN1350
      WRITE (6,52) CNEQ,NEQ,CNPAR,NPAR ,CNT,NT,CTF,TE,CDTZ,DTZ,CDOUT,DOUMAIN1360
     1TZ,CIOP,IOP                                                         MAIN1370
      IF((NEQ.LT.1) .OR. (NEQ.GT.150)) GO TO 195                          MAIN1380
      IF((NPAR.LT.0).OR.(NPAR.GT.100)) GO TO 195                          MAIN1390
      IF((IOP.LT.1) .OR. (IOP.GT.4)) GO TO 195                            MAIN1400
      IF(DTZ.GT.DOUTZ) GO TO 195                                          MAIN1410
      GO TO(110,111,111,112),IOP                                          MAIN1420
  110 WRITE (6,68)                                                        MAIN1430
      GO TO 120                                                           MAIN1440
  111 WRITE (6,69)                                                        MAIN1450
      GO TO 120                                                           MAIN1460
  112 WRITE (6,70)                                                        MAIN1470
C                                                                         MAIN1480
C     ADDITIONAL DATA FOR VARIABLE STEP INTEGRATION OPTION                MAIN1490
C                                                                         MAIN1500
  120 GO TO (130,123,124,130),IOP                                         MAIN1510
  123 READ (5,3) CE,E,CHMIN,HMIN,CHMAX,HMAX,CBETA,BETA,CA,A,CF,F          MAIN1520
      WRITE (6,64) CE,E,CHMIN,HMIN,CHMAX,HMAX,CBETA,BETA,CA,A,CF,F        MAIN1530
      IF (F.LT. 2.) GO TO 195                                             MAIN1540
      IF(HMAX.LE.HMIN) GO TO 195                                          MAIN1550
      IF(BETA.GE.1.) GO TO 195                                            MAIN1560
      GO TO 130                                                           MAIN1570
  124 IOP1=IOP-2                                                          MAIN1580
      E=1.E-05                                                            MAIN1590
      HMIN=1.E-05                                                         MAIN1600
      HMAX=100.                                                           MAIN1610
      BETA=0.5                                                            MAIN1620
      A=1.                                                                MAIN1630
      F=100.                                                              MAIN1640
      WRITE (6,65) E,HMIN,HMAX,BETA,A,F                                   MAIN1650
  130 IF(NT.EQ.0) GO TO 196                                               MAIN1660
      IF(NT.GT.MM) GO TO 194                                              MAIN1670
C                                                                         MAIN1680
C     READ AND WRITE TABLES                                               MAIN1690
C                                                                         MAIN1700
  150 CONTINUE                                                            MAIN1710
      DO 180 I=1,NT                                                       MAIN1720
      READ (5,2) CNP,NP(I)                                                MAIN1730
      NN=NP(I)                                                            MAIN1740
      WRITE (6,55) CNP,I,NN                                               MAIN1750
      IF(NN-MMM)170,170,160                                               MAIN1760
  160 WRITE (6,56)                                                        MAIN1770
      GO TO 195                                                           MAIN1780
```

```
  170 CONTINUE
      READ (5,4) (  T1(K,I),  T2(K,I),K=1,NN)
      WRITE (6,62) (  T1(K,I),  T2(K,I),K=1,NN)
  180 CONTINUE
      GO TO 196
C
  194 WRITE (6,54) MM
  195 WRITE (6,66)
      STOP
C
  196 DO 197 I=6,10
  197 S(I)=.FALSE.
      DO 198 K=1,NEQ
  198 X(K)=XZ(K)
      ILM=NT+1
      DO 199 I=ILM,MM
  199 NP(I)=0
      NOREP=.FALSE.
      KIT=1
C
C
C                                                   **********************
C                                                   *                    *
C                                                   * RETURN AFTER REPEAT *
C                                                   *                    *
C                                                   **********************
  200 CONTINUE
C     INITIALISATION FOR CALCOMP ROUTINE (CURVE)
      N1=NEQ
      N2=0
      NA=0
      NAXY=0
      KA=0
      NCALL=0
      NCALLX=0
      DO 201 I=1,15
  201 NCURV(I)=0
      DO 202 I=1,6
  202 NCURVX(I)=0
      DO 203 I=1,150
  203 NS(I)=I
      NOCALC=.TRUE.
      NOXY=.TRUE.
C
      BEGIN=.TRUE.
```

```
      DO 210 I= 1 ,MM                                                   MAIN2240
  210 ATT(I)=0.                                                         MAIN2250
      T=0.0                                                             MAIN2260
      DT=DTZ                                                            MAIN2270
      DOUT=DOUTZ                                                        MAIN2280
      OUT=DOUTZ*0.99999                                                 MAIN2290
      TF=TE                                                             MAIN2291
      TSTOP=TF                                                          MAIN2300
      TOLD=T                                                            MAIN2310
      DO 216 I=1,5                                                      MAIN2320
  216 S(I)=.FALSE.                                                      MAIN2330
      S(12)=.FALSE.                                                     MAIN2340
      DO 217 I=13,15                                                    MAIN2350
  217 S(I)=.TRUE.                                                       MAIN2360
      DO 221 I=1,20                                                     MAIN2370
  221 SL(I)=.FALSE.                                                     MAIN2380
      RESET=DT                                                          MAIN2390
C     WHEN CALLED SWITCH RESETS S(I) TO TRUE                            MAIN2400
C                                                                       MAIN2410
C     COMPUTE INITIAL DERIVATIVES                                       MAIN2420
C     ***************************************************************** MAIN2430
      CALL DER(T,X,DX,PAR)                                              MAIN2440
C     ***************************************************************** MAIN2450
C                                                                       MAIN2460
  222 IF(KIT.GT.1) GO TO 290                                            MAIN2470
      IZ=1                                                              MAIN2480
      I30=30                                                            MAIN2490
      NEQ1=NEQ                                                          MAIN2500
  230 IF(NEQ1.LE.30) GO TO 235                                          MAIN2510
      WRITE (6,57)    ((CXZ(K,I),K=1,10),I,XZ(I),I,DX(I),I=IZ,I30)      MAIN2520
      IZ=I30+1                                                          MAIN2530
      NEQ1=NEQ-I30                                                      MAIN2540
      I30=I30+30                                                        MAIN2550
      GO TO 230                                                         MAIN2560
  235 WRITE (6,57)    ((CXZ(K,I),K=1,10),I,XZ(I),I,DX(I),I=IZ,NEQ)      MAIN2570
  240 NST=3+N1                                                          MAIN2580
C                                 NST=N. OF LINES PER OUTPUT            MAIN2590
      IF(N2)270,270,250                                                 MAIN2600
  250 ID=1                                                              MAIN2610
      N3=N2+1                                                           MAIN2620
      IF(N1-N2)280,260,290                                              MAIN2630
  260 ID=4                                                              MAIN2640
      N3=1                                                              MAIN2650
      GO TO 290                                                         MAIN2660
```

```
  270 ID=2
      N3=1
      GO TO 290
  280 ID=3
      N3=N1+1
      NST=3+N2
  290 CONTINUE
      NL=NST
      IF(NPAR.EQ.0) GO TO 295
      WRITE (6,53) ((CPAR(K,I),K=1,10),I,PAR(I),I=1,NPAR)
  295 WRITE (6,58)
C     READ OUT AT T=0
      CALL WRPP
C
C
C                                   ********************
C                                   *                  *
C                                   * INTEGRATION LOOP *
C                                   *                  *
C                                   ********************
C
C     MODIFY INTEGRATION (FIXED) OR OUTPUT TIME STEP
  300 CONTINUE
      GO TO (310,330,330,310),IOP
C       SWITCH(13) MODIFIES TIME STEP
  310 IF(SWITCH(13)) GO TO 330
      GO TO (312,320,320,320),IOP
  312 S(15)= .FALSE.
C     PREDICTOR CORRECTOR IS STARTED AGAIN BY SWITCH(15)
  320 DT=FMT
C       SWITCH(14) MODIFIES OUTPUT STEP
  330 IF(SWITCH(14)) GO TO 350
      OUT=OUT-DOUT
      DOUT=FMX
      OUT=OUT+DOUT
  350 CONTINUE
      IF(TOLD.LT.T) TOLD=T
      DTOLD=DT
C
C     INTEGRATION ROUTINE SELECTION DEPENDING ON IOP
C     *****************************************************************
C            IOP=1,PREDICTOR-CORRECTOR,FIXED STEP
C            IOP=2,PREDICTOR-CORRECTOR,VARIABLE STEP
C            IOP=3,PREDICTOR-CORRECTOR,VARIABLE STEP,STANDARD DATA
C            IOP=4,4TH ORDER RUNGE-KUTTA,FIXED STEP
```

```
C
      GO TO (351,351,351,354),IOP
C     SWITCH (15) STARTS AGAIN PREDICTOR-CORRECTOR BY RUNGE-KUTTA
  351 IF(SWITCH(15)) GO TO 353
      IENTRY=0
      WRITE(6,67) T
  353 CALL NADM(IENTRY)
      IENTRY=1
      GO TO 360
  354 CALL NRM1
C
C     TEST FOR FINAL TIME OR START TIME
  360 IF(T+DTOLD-TSTOP)375,361,361
  361 IF(T-TSTOP)363,363,362
  362 T=T-DT
  363 DT=TSTOP-T
C
C     LAST INTEGRATION STEP BEFORE TSTOP USES RUNGE-KUTTA
C     ******************************************************************
      CALL NRM1
C
      IF(TSTOP-TF)370,365,365
  365 BEGIN=.FALSE.
  370 S(15)=.FALSE.
      TSTOP=TF
      DT=DTOLD
      OUT=T+0.99999
C
C
C     CARRY ON WITH INTEGRATION
  375 IF(T-TOLD)300,300,376
  376 IF(T-OUT)300,380,380
  380 OUT=OUT+DOUT
      NL=NL+NST
      IF(NL-55)400,400,390
  390 NL=NST
      WRITE (6,58)
C     READ OUT
  400 CALL WRPP
      IF(BEGIN) GO TO 300
C                           ****************************************
C                           *                                      *
C                           * END OF INTEGRATION AND READ OUT LOOP *
C                           *                                      *
```

```
C                               ****************************************     MAIN3560
C                                                                            MAIN3570
C     CARRY ON FOR REPEAT LOOP                                               MAIN3580
C                                                                            MAIN3590
      IF(NOCALC) GO TO 430                                                   MAIN3600
      NCXY=NAXY-NCALLX                                                       MAIN3610
  420 WRITE (6,59) NA,NCXY,KA                                                MAIN3620
C     NA=N.OF TIME CURVES,NCXY=N.OF  X-Y CURVES,KA=N.OF POINTS PER CURVEMAIN3630
      CALL CURVE                                                             MAIN3640
  430 CALL REPEAT(X,XZ,PAR)                                                  MAIN3650
      IF(NOREP) GO TO 100                                                    MAIN3660
  440 WRITE (6,60) KIT                                                       MAIN3670
      KIT=KIT+1                                                              MAIN3680
      IF(KIT.LT.26)  GO TO 200                                               MAIN3690
  450 WRITE (6,61)                                                           MAIN3700
      GO TO 100                                                              MAIN3710
      END                                                                    MAIN3720
```

```
      SUBROUTINE NRM1
C*****CONSTANT STEP 4 TH  ORDER  RUNGE KUTTA INTEGRATION***************
C     USED ALSO TO START NADM (PRED.CORR.VAR.STEP)
      COMMON/VARIAB/N,X,Y(150),DY(150),DX,E,F,HMIN,HMAX,BETA,A,IOPT
      COMMON/RAP/PAR(100)
      DIMENSION VK(150,3),Z(150),HH(2)
      HH(1)=DX/2.
      HH(2)=DX
      DO 10 I=1,N
   10 Z(I)=Y(I)+HH(1)*DY(I)
C
      CALL DER(X+HH(1),Z,VK(1,1),PAR)
C
      DO 25 M=1,2
      DO 20 I=1,N
   20 Z(I)=Y(I)+HH(M)*VK(I,M)
C
      CALL DER(X+HH(M),Z,VK(1,M+1),PAR)
C
   25 CONTINUE
      DO 40 I=1,N
   40 Y(I)=Y(I)+HH(2)/6.*(DY(I)+2.*(VK(I,1)+VK(I,2))+VK(I,3))
      X=X+HH(2)
C
      CALL DER(X,Y,DY,PAR)
C
      RETURN
      END
```

```
      SUBROUTINE NADM(NENTRY)
C
C*****ADAMS-MOULTON PREDICTOR CORRECTOR,FIXED OR VARIABLE STEP*********
C
      COMMON/VARIAB/N,X,Y(150),ADY(150),DX,E,F,HMIN,HMAX,BETA,A,IOPT
      COMMON/RAP/PAR(100)
      DIMENSION YBEGIN(150),YCOR(150),YPRED(150),DY(150,3),BDY(150)
C     PREDICTOR-CORRECTOR FORTRAN IV
  500 FORMAT(1H0,6X,41HWARNING MINIMUM TIME STEP USED AT TIME =
     1,1PE12.5/)
      IF(NENTRY)101,102,101
C     INITIALISATION ENTRY
  102 CONTINUE
      EDF=E/F
  303 CONTINUE
      DO 1 I=1,N
      DY(I,3)=ADY(I)
    1 YBEGIN(I)=Y(I)
   30 CONTINUE
      ISTEP=0
      KOUNT=0
      RETURN
C
C
C     INTEGRATION ENTRY
  101 CONTINUE
      IF(ISTEP.GE.3) GO TO 2
      AMAX=0.
      CALL NRM1
      K=2-ISTEP
      IF(K)42,42,41
   41 CONTINUE
      DO 8 I=1,N
    8 DY(I,K)=ADY(I)
   42 CONTINUE
      ISTEP=ISTEP+1
      RETURN
C
C     START PRED-COR
    2 CONTINUE
C     CALCULATE PREDICTED Y
      AX=DX/24.
      DO 5 I=1,N
    5 YPRED(I)=Y(I)+AX*(55.*ADY(I)-59.*DY(I,1)+37.*DY(I,2)-9.*DY(I,3))
      X=X+DX
```

```
C                                                                         NADM0460
      CALL DER(X,YPRED, BDY,PAR)                                          NADM0470
C                                                                         NADM0480
C                                                                         NADM0490
C     CALCULATE CORRECTED Y                                               NADM0500
      DO 7 I=1,N                                                          NADM0510
    7 YCOR(I)=Y(I)+AX*(9.* BDY(I)+19.*ADY(I)-5.*DY(I,1)+DY(I,2))          NADM0520
      AMAX=0.                                                             NADM0530
      DO 9 I=1,N                                                          NADM0540
      D=ABS(YCOR(I))                                                      NADM0550
      IF(A-D)71,71,72                                                     NADM0560
   72 CONTINUE                                                            NADM0570
      D=A                                                                 NADM0580
   71 CONTINUE                                                            NADM0590
      ERR=(ABS(YCOR(I)-YPRED(I)))/D                                       NADM0600
      IF(ERR-AMAX)9,9,11                                                  NADM0610
   11 AMAX=ERR                                                            NADM0620
    9 CONTINUE                                                            NADM0630
      AMAX=AMAX/14.                                                       NADM0640
      IF(IOPT)10,12,10                                                    NADM0650
C                                                                         NADM0660
C     VARIABLE STEP                                                       NADM0670
   10 CONTINUE                                                            NADM0680
      IF(AMAX-E)13,14,14                                                  NADM0690
   13 CONTINUE                                                            NADM0700
      IF(AMAX-EDF)15,45,45                                                NADM0710
   14 CONTINUE                                                            NADM0720
      IF(DX*BETA-HMIN)44,45,16                                            NADM0730
   15 CONTINUE                                                            NADM0740
      IF(HMAX)19,20,19                                                    NADM0750
   19 CONTINUE                                                            NADM0760
C     DX ALREADY=HMAX                                                     NADM0770
      IF(DX/BETA-HMAX)20,12,12                                            NADM0780
   20 CONTINUE                                                            NADM0790
C     TEST IF THIRD STEP WITH LOW ERROR                                   NADM0800
      KOUNT=KOUNT+1                                                       NADM0810
      IF(KOUNT-3)12,21,12                                                 NADM0820
   44 WRITE(6,500)  X                                                     NADM0830
   45 KOUNT=0                                                             NADM0840
C                                                                         NADM0850
   12 CONTINUE                                                            NADM0860
C     CARRY ON WITH PRED-COR.                                             NADM0870
      DO 50 I=1,N                                                         NADM0880
      DY(I,3)=DY(I,2)                                                     NADM0890
```

```
      DY(I,2)=DY(I,1)
      DY(I,1)=ADY(I)
      Y(I)=YCOR(I)
   50 CONTINUE
      ISTEP=ISTEP+1
C
      CALL DER(X,YCOR,ADY,PAR)
C
      RETURN
C
   21 CONTINUE
C     INCREASE STEP,START WITH RK FROM THIS POINT
      DX=DX/BETA
      DO 59 I=1,N
      Y(I)=YCOR(I)
   59 CONTINUE
C
      CALL DER(X,Y,ADY,PAR)
C
      GO TO 303
C
   18 CONTINUE
C     DECREASE STEP
      IF(ISTEP-3)52,53,52
C
   53 CONTINUE
C     RETURN 3 STEPS,TO RESTART RK
      X=X-4.*DX
      DX=DX*BETA
      DO 54 I=1,N
      Y(I)=YBEGIN(I)
      ADY(I)=DY(I,3)
   54 CONTINUE
      GO TO 30
C
   52 CONTINUE
C     RETURN 1 STEP,TO RESTART WITH RK
      X=X-DX
      DX=DX*BETA
      GO TO 303
C
      END
```

```
      FUNCTION SIGMA(X,I1,I2)
C     MULTIPLE SUMMATION FROM I1 TO I2.
      DIMENSION X(30)
      A=0.
      DO 11 I=I1,I2
      A=X(I)+A
   11 CONTINUE
      SIGMA=A
      RETURN
      END
```

```
      FUNCTION SIGMAP(X,Y,I1,I2)                                        SAHY0130
C     MULTIPLE SUMMATION OF PRODUCT FROM I1 TO I2                       SAHY0140
      DIMENSION X(30),Y(30)                                             SAHY0150
      A=0.                                                              SAHY0160
      DO 10 I=I1,I2                                                     SAHY0170
      A=X(I)*Y(I)+A                                                     SAHY0180
   10 CONTINUE                                                          SAHY0190
      SIGMAP=A                                                          SAHY0200
      RETURN                                                            SAHY0210
      END                                                               SAHY0220
```

```
      LOGICAL FUNCTION SWITCH(I)                                    SAHY0240
      COMMON/HCTIWS/S(15)                                           SAHY0250
      LOGICAL S                                                     SAHY0260
C                                                                   SAHY0270
C     SWITCH 1  TO 5  USED BY DER                                   SAHY0280
C     SWITCH 6  TO 10 USED BY REPEAT                                SAHY0290
C     SWITCH 12 USED BY FTWOV                                       SAHY0300
C     S(13) IS USED BY MODDT                                        SAHY0310
C     S(14) IS USED BY MODDX                                        SAHY0320
C     S(15) IS USED BY MAIN TO START PREDICTOR-CORRECTOR            SAHY0330
C                                                                   SAHY0340
      SWITCH=S(I)                                                   SAHY0350
      S(I)=.TRUE.                                                   SAHY0360
      RETURN                                                        SAHY0370
      END                                                           SAHY0380
```

```
      SUBROUTINE MODDX(GAM)                                             SAHY0400
      COMMON/HCTIWS/S(15)                                               SAHY0410
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD                         SAHY0420
      LOGICAL S                                                         SAHY0430
C                                                                       SAHY0440
C     MODIFIES READ OUT STEP                                            SAHY0450
C     MUST BE CALLED BY DER                                             SAHY0460
C                                                                       SAHY0470
      IF(GAM.EQ.GAMOLD) GO TO 100                                       SAHY0480
      GAMOLD=GAM                                                        SAHY0490
      S(14)=.FALSE.                                                     SAHY0500
      FMX=GAM                                                           SAHY0510
  100 RETURN                                                            SAHY0520
      END                                                               SAHY0530
```

```
      SUBROUTINE MODDT(GAM)
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD
      COMMON/HCTIWS/S(15)
      LOGICAL S
C
C        MODIFIES TIME STEP
C     MUST BE CALLED BY DER
C
      IF(GAM.EQ.GAMOLD) GO TO 100
      GAMOLD=GAM
      S(13)=.FALSE.
      FMT=GAM
  100 RETURN
      END
```

```
      SUBROUTINE START(TIME)
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1
C
C     ENDS CURRENT INTEGRATION RUN AT TIME T AND FROM THERE STARTS
C     AGAIN PREDICTOR CORRECTOR WITH RUNGE KUTTA
C
      IF(T+DT-TIME)30,10,10
   10 IF(RESET-TIME)25,30,30
   25 TSTOP=TIME
      RESET=TIME+DTOLD
   30 RETURN
      END
```

```
      SUBROUTINE STOP(TIME)                                              SAHY0840
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1    SAHY0850
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD                          SAHY0860
      IF(T+DT-TIME)20,10,10                                              SAHY0870
   10 TSTOP=TIME                                                         SAHY0880
      TF=TIME                                                            SAHY0890
   20 RETURN                                                             SAHY0900
      END                                                                SAHY0910
```

```
      SUBROUTINE STOPR                                                  SAHY0930
      COMMON/REP/NOREP                                                  SAHY0940
      LOGICAL NOREP                                                     SAHY0950
      NOREP=.TRUE.                                                      SAHY0960
      RETURN                                                            SAHY0970
      END                                                               SAHY0980
```

```
      FUNCTION PULSE(TIME)                                              SAHY1000
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD                         SAHY1010
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1   SAHY1020
C     UNIT SURFACE PULSE AND INTEGRATION STEP ADJUSTMENT                SAHY1030
      IF(T+DT-TIME)10,5,5                                               SAHY1040
    5 IF(T-DT-DT-TIME)20,10,10                                          SAHY1050
   10 PULSE=0.0                                                         SAHY1060
      GO TO 100                                                         SAHY1070
   20 CALL START(TIME)                                                  SAHY1080
      IF(T-TIME)10,30,40                                                SAHY1090
   30 PULSE=1./DTOLD                                                    SAHY1100
      GO TO 100                                                         SAHY1110
   40 PULSE=0.0                                                         SAHY1120
      CALL START(TIME+1.000001*DT)                                      SAHY1130
  100 RETURN                                                            SAHY1140
      END                                                               SAHY1150
```

```
      FUNCTION STEP (TIME)                                              SAHY1170
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1   SAHY1180
C     UNIT STEP FUNCTION AND INTEGRATION STEP ADJUSTMENT                SAHY1190
      IF(T+DT-TIME)10,20,20                                             SAHY1200
   10 STEP=0.0                                                          SAHY1210
      GO TO 40                                                          SAHY1220
   20 CALL START(TIME)                                                  SAHY1230
      IF(T-TIME)10,30,30                                                SAHY1240
   30 STEP=1.                                                           SAHY1250
   40 RETURN                                                            SAHY1260
      END                                                               SAHY1270
```

```
      FUNCTION RAMP(TIME1,TIME2)                                        SAHY1290
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1   SAHY1300
C     RAMP FUNCTION AND INTEGRATION STEP ADJUSTMENT                     SAHY1310
      IF(T+DT-TIME1)10,20,20                                            SAHY1320
   10 RAMP=0.0                                                          SAHY1330
      GO TO 100                                                         SAHY1340
   20 CALL START(TIME1)                                                 SAHY1350
      IF(T-TIME1)10,30,30                                               SAHY1360
   30 IF(T+DT-TIME2)40,50,50                                            SAHY1370
   40 RAMP=T-TIME1                                                      SAHY1380
      GO TO 100                                                         SAHY1390
   50 CALL START(TIME2)                                                 SAHY1400
      IF(T-TIME2)40,60,60                                               SAHY1410
   60 RAMP=TIME2-TIME1                                                  SAHY1420
  100 RETURN                                                            SAHY1430
      END                                                               SAHY1440
```

```
      LOGICAL FUNCTION SWLOG(I)
C     USED BY COMPAR
      LOGICAL SL
      COMMON/GOLWS/SL(20)
      SWLOG=SL(I)
      SL(I)=.TRUE.
      RETURN
      END
```

```
      FUNCTION COMPAR(Z,I)
C     COMPARATOR AND PREDICTOR CORRECTOR START
      LOGICAL SL,SWLOG
      COMMON/GOLWS/SL(20)
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1
      COMMON/TRATS/FMX,FMT,TSTOP,RESET,TF,DTOLD
      COMMON/OLDVAL/TOLD
    1 FORMAT (1H0///24HCOMPARATOR INDEX EXCEEDS,I3,12HNO EXECUTION)
    2 FORMAT(1H0,6X,11HCOMPARATOR(,I2,3H) =,F2.0,10H AT TIME =,1PE12.5)
C
      DATA NCOMP/10/
C
C     NCOMP  IS THE NUMBER OF COMPAR.AVAILABLE
C     THE DIMENSION OF  SL  MUST BE 2*NCOMP
C
      IF(I.LE.NCOMP) GO TO 9
      WRITE (6,1) NCOMP
      STOP
    9 II=I+NCOMP
      IF(Z)10,20,20
   10 COMPAR=0.0
      IF(SWLOG(I)) GO TO 100
      SL(II)=.FALSE.
      GO TO 30
C
   20 COMPAR=1.0
      IF(SWLOG(II)) GO TO 100
      SL(I)=.FALSE.
C
   30 IF(T.LE.DT) GO TO 100
      TSTOP=TOLD+DTOLD
      WRITE(6,2) I,COMPAR,T
  100 RETURN
      END
```

```
      SUBROUTINE STAMPA
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1
      COMMON/UTIL/NS(150),N1,N2,N3,ID,VN(50),VE(50),CXZ(10,150)
      COMMON/CALCOM/TABTX(128,16),HCURV(15),NCALL,NCURV(15),TABXY(128,6)
     1,HCURVX(6),NCALLX,NCURVX(6),KA,NA,NAXY,NOCALC,HEAD(18),NOXY
      LOGICAL NOCALC,NOXY
    1 FORMAT (1H0, 6X,7HTIME = ,1PE12.5,17X,12HTIME STEP = ,1PE12.5//)
    2 FORMAT (1H , 6X,2HX(,I3,4H) = ,A4,A2,3H = ,1PE12.5,6X,4HDER(,A4,A2
     1,4H) = ,1PE12.5,6X,A6,3H = ,1PE12.5)
    3 FORMAT (1H , 6X,2HX(,I3,4H) = ,A4,A2,3H = ,1PE12.5,6X,4HDER(,A4,A2
     1,4H) = ,1PE12.5)
    4 FORMAT (1H , 74X,A6,3H = ,1PE12.5)
C     NOXY=FALSE IF DRAWXY CALLED
      IF(KA-128)101,200,200
  101 KA=KA+1
      IF(NOCALC) GO TO 110
      CALL DRAW1(TABTX(KA,1))
  110 IF(NOXY) GO TO 200
      CALL DRAW2(TABXY(KA,1))
C                   **N1=N. INTEGRATED VAR.**
C                   **N2=N. AUXILIARY VAR.**
  200 WRITE (6,1) T,DT
      GO TO(210,220,230,240),ID
  210 DO 211 I=1,N2
      I1=NS(I)
  211 WRITE (6,2) I1,CXZ(1,I1),CXZ(2,I1),X(I1),CXZ(1,I1),CXZ(2,I1),
     1DX(I1),VN(I),VE(I)
  220 DO 221 I=N3,N1
      I1=NS(I)
  221 WRITE (6,2) I1,CXZ(1,I1),CXZ(2,I1),X(I1),CXZ(1,I1),CXZ(2,I1),
     1DX(I1)
  222 RETURN
  230 DO 231 I=1,N1
      I1=NS(I)
  231 WRITE (6,2) I1,CXZ(1,I1),CXZ(2,I1),X(I1),CXZ(1,I1),CXZ(2,I1),
     1DX(I1),VN(I),VE(I)
  232 WRITE (6,4) (VN(I),VE(I),I=N3,N2)
      GO TO 222
  240 DO 241 I=1,N1
      I1=NS(I)
  241 WRITE (6,2) I1,CXZ(1,I1),CXZ(2,I1),X(I1),CXZ(1,I1),CXZ(2,I1),
     1DX(I1),VN(I),VE(I)
      GO TO 222
      END
```

```
      SUBROUTINE STAMP1(X,Y,N)                                          STAM0460
      DIMENSION X(10),Y(10)                                             STAM0470
      WRITE (6,1) (X(I),Y(I),I=1,N)                                     STAM0480
    1 FORMAT (1H1,10X,36HPARAMETERS COMPUTED IN FIRST SECTION///(11X,A6,STAM0490
     13H = ,1PE13.5))                                                   STAM0500
      RETURN                                                            STAM0510
      END                                                               STAM0520
```

```
      SUBROUTINE CURVE
      COMMON/CALCOM/TABTX(128,16),HCURV(15),NCALL,NCURV(15),TABXY(128,6)
     1,HCURVX(6),NCALLX,NCURVX(6),KA,NA,NAXY,NOCALC,HEAD(18),NOXY
      DIMENSION CAR(15)
      DATA  CAR            /1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,2H10,2H11
     1,2H12,2H13,2H14,2H15/
    1 FORMAT (1H1)
    2 FORMAT (1H0//9X,23H***********PLOT NUMBER ,I3, 9X,I3,3X,6HPOINTS/
     1/21X,20HTIME , FULL SCALE = 1PE13.2)
    3 FORMAT (1H0,20X,A6,5X,5HCURVE,I3,5X,23HZERO AMPLITUDE NO PLOT /)
    4 FORMAT (1H0,8X,5HCURVE,I3,2X,A6,16H , FULL SCALE = ,1PE13.2)
    5 FORMAT (1H0,20X,17HX Y PLOT NUMBER , I3,10X,A6,5X,24HZERO AMPLITUD
     1E , NO PLOT/)
    6 FORMAT (1H0//9X,27H***********X Y PLOT NUMBER ,I3, 5X,I3,3X,6HPOI
     1NTS// 9X,8HABSCISSA ,2X,A6,16H , FULL SCALE = ,1PE13.2)
    7 FORMAT (1H0,20X,33H**FINAL TIME EQUAL ZERO.NO PLOT**)
      LOGICAL NOCALC,NOXY
      FST=40.
      FSX=20.
      FSY=20.
C     FST=MAX LENGTH OF TIME AXIS
C     FSY=MAX LENGTH OF Y AXIS
C     FSX=MAX LENGTH OF X AXIS
      ZEROP=1.E-37
C     NO PLOT OF CURVE IF ABS AMPLITUDE BELOW ZEROP
      FSY2=2.*FSY
      FSX2=2.*FSX
      KAM=KA/4
      KR=KA-KAM+1
C                                **TIME PLOTS**
      IF(NCALL)210,210,10
   10 I1=0
C                                **TIME SCALE**
      YMM=TABTX(KA,16)
      IF(YMM-ZEROP)11,11,12
   11 WRITE(6,7)
      STOP
   12 CONTINUE
      CALL SCALA(YMM,FST,TABTX(1,16))
      DO 90 IN=1,NCALL
      I2=I1+1
      I3=I1+NCURV(IN)
      WRITE (6,2) IN,KA,YMM
C                                **Y SCALE**
      DO 30 IK=I2,I3
```

```
      YM=VALMAX(TABTX(1,IK))
      IF(YM-ZEROP)20,20,25
   20 WRITE (6,3) HCURV(IK),IK
      GO TO 30
   25 CALL SCALA(YM,FSY,TABTX(1,IK))
      WRITE (6,4) IK,HCURV(IK),YM
   30 CONTINUE
C                                 **ACTUAL PLOT**
   60 CALL FINIM(2.,15.)
      CALL PLOT(0.,FSY2,2)
      CALL PLOT(FST,FSY,3)
      CALL PLOT(0.,FSY,2)
      CALL SYMBL4(-1.5,-1.,0.28,0.,CAR(IN),2,0)
      CALL SYMBL4(0.,-1.,0.28,0.,HEAD,24,0)
      CALL FINIM(0.,FSY)
      DO 70 I=I2,I3,2
      CALL LINE(TABTX(1,16),TABTX(1,I),KAM,1,1)
      CALL SYMBL4(TABTX(KAM,16),TABTX(KAM,I)+0.2,0.28,0.,CAR(I),2,0)
      CALL LINE(TABTX(KAM,16),TABTX(KAM,I),KR,1,1)
      IP=I+1
      IF(IP-I3)65,65,80
   65 CALL LINE(TABTX(KA,16),TABTX(KA,IP),KR,-1,-1)
      CALL SYMBL4(TABTX(KAM,16),TABTX(KAM,IP),0.28,0.,CAR(IP),2,0)
      CALL LINE(TABTX(KAM,16),TABTX(KAM,IP),KAM,-1,-1)
   70 CONTINUE
   80 CALL FINIM(FST+2.,-FSY-15.)
   90 I1=I3
C     ****************************************************************
C                                 **X Y PLOTS**
C                                 **X SCALE**
  210 IF(NCALLX)500,500,220
  220 I1=0
      N=0
      WRITE (6,1)
      DO 320 IN=1,NCALLX
      N=N+1
      I2=I1+1
      I3=I2+1
      YM=VALMAX(TABXY(1,I2))
      I4=I1+NCURVX(IN)
      IF(YM-ZEROP)230,230,240
  230 WRITE (6,5) IN,HCURVX(I2)
      GO TO 310
  240 CALL SCALA (YM,FSX,TABXY(1,I2))
```

```
      WRITE (6,6) IN,KA,HCURVX(I2),YM                                   CURV0890
C                                        **Y SCALE**                    CURV0900
      DO 260 IK=I3,I4                                                   CURV0910
      YM=VALMAX(TABXY(1,IK))                                            CURV0920
      IF(YM-ZEROP)245,245,250                                           CURV0930
  245 WRITE (6,5) IN,HCURVX(IK)                                         CURV0940
      GO TO 260                                                         CURV0950
  250 CALL SCALA(YM,FSY,TABXY(1,IK))                                    CURV0960
      IKN=IK-N                                                          CURV0970
      WRITE (6,4) IKN,HCURVX(IK),YM                                     CURV0980
  260 CONTINUE                                                          CURV0990
C                                        **ACTUAL PLOT**                CURV1000
  270 CALL FINIM(2.+FSX,15.)                                            CURV1010
      CALL PLOT(0.,FSY2,2)                                              CURV1020
      CALL PLOT(FSX,FSY,3)                                              CURV1030
      CALL PLOT(-FSX,FSY,2)                                             CURV1040
      CALL SYMBL4(-1.5,-1.,0.28,0.,CAR(IN),2,0)                         CURV1050
      CALL SYMBL4(0.,-1.,0.28,0.,HEAD,24,0)                             CURV1060
      CALL FINIM( 0. ,FSY)                                              CURV1070
      DO 290 IK=I3,I4,2                                                 CURV1080
      IKN=IK-N                                                          CURV1090
      CALL LINE(TABXY(1,I2),TABXY(1,IK),KAM,1,1)                        CURV1100
      CALL SYMBL4(TABXY(KAM,I2),TABXY(KAM,IK) +0.2,0.28,0.,CAR(IKN),2,0)CURV1110
      CALL LINE(TABXY(KAM,I2),TABXY(KAM,IK),KR,1,1)                     CURV1120
      IR=IK+1                                                           CURV1130
      IF(IR-I4)280,280,300                                              CURV1140
  280 CALL LINE(TABXY(KA,I2),TABXY(KA,IR),KR,-1,-1)                     CURV1150
      IKN=IKN+1                                                         CURV1160
      CALL SYMBL4(TABXY(KAM,I2),TABXY(KAM,IR),0.28,0.,CAR(IKN),2,0)     CURV1170
      CALL LINE(TABXY(KAM,I2),TABXY(KAM,IR),KAM,-1,-1)                  CURV1180
  290 CONTINUE                                                          CURV1190
  300 CALL FINIM(FSX +2.,-FSY-15.)                                      CURV1200
  310 I1=I4                                                             CURV1210
  320 CONTINUE                                                          CURV1220
  500 RETURN                                                            CURV1230
      END                                                               CURV1240
```

```
      SUBROUTINE SCALA(YM,FS,Y)
C     SCALE FACTOR FOR CALCOMP PLOTS  IN PHYS UNITS PER CM
      COMMON/CALCOM/TABTX(128,16),HCURV(15),NCALL,NCURV(15),TABXY(128,6)
     1,HCURVX(6),NCALLX,NCURVX(6),KA,NA,NAXY,NOCALC,HEAD(18),NOXY
      DIMENSION Y(128)
C     YM=MAX AMPLITUDE FS=FULL SCALE IN CM
      KK=1
      AK=10.
      AKM=0.1
      IF(YM-1.)200,100,100
  100 IF(YM-AK)102,102,101
  101 KK=KK+1
      AK=AK*10.
      GO TO 100
  102 KK=KK-1
      GO TO 300
  200 IF(YM-AKM)201,202,202
  201 KK=KK+1
      AKM=AKM*0.1
      GO TO 200
  202 KK=-KK
  300 CONTINUE
      YM=0.99*YM
      MY=(YM*(10.0**(-KK)))
      MY=MY+1
      YM=FLOAT(MY)*(10.0**KK)
      SCA=YM/FS
      DO 400 I=1,KA
  400 Y(I)=Y(I)/SCA
      RETURN
      END
```

```
      FUNCTION VALMAX(AY)
C     SEARCH FOR MAXIMUM ABSOLUTE AMPLITUDE
      COMMON/CALCOM/TABTX(128,16),HCURV(15),NCALL,NCURV(15),TABXY(128,6)
     1,HCURVX(6),NCALLX,NCURVX(6),KA,NA,NAXY,NOCALC,HEAD(18),NOXY
      DIMENSION AY(128)
      YM=0.0
      DO 10 J=1,KA
      YA=ABS(AY(J))
      IF(YM-YA)5,10,10
    5 YM=YA
   10 CONTINUE
      VALMAX=YM
      RETURN
      END
```

```
      FUNCTION FTWOV(I1,I2,X,Z,VZ)                                      FTW00010
      COMMON/HCTIWS/S(15)                                               FTW00020
      LOGICAL SWITCH                                                    FTW00030
      DIMENSION VZ(30)                                                  FTW00040
C                                                                       FTW00050
    1 FORMAT (1H0/1H0,6X,6H** Z =,1PE12.5,52HOUTSIDE RANGE OF FUNCTION OFTW00060
     1F 2 VAR.DEFINED BY TABLES,I3,3HTO ,I3//)                          FTW00070
    2 FORMAT (1H0/1H0,6X,41HWRONG ARGUMENT IN FTWOV DEFINED BY TABLES ,IFTW00080
     13,3H TO ,I3//)                                                    FTW00480
C                                                                       FTW00490
      XX=X                                                              FTW00500
      NN=I2-I1+1                                                        FTW00510
      IF (SWITCH(12)) GO TO 5                                           FTW00520
      IF (I1.GE.I2.OR.I2.GT.30) GO TO 210                               FTW00530
      DO 4 I=2,NN                                                       FTW00540
      IF(VZ(I-1).GE.VZ(I)) GO TO 210                                    FTW00550
    4 CONTINUE                                                          FTW00560
C                                                                       FTW00570
    5 IF((Z.LT.VZ(1)).OR.(Z.GT.VZ(NN))) GO TO 200                       FTW00580
      DO 10 I=1,NN                                                      FTW00590
      IF(Z-VZ(I))30,20,10                                               FTW00600
   10 CONTINUE                                                          FTW00610
   20 FTWOV=TABLE(I,XX)                                                 FTW00620
      GO TO 100                                                         FTW00630
   30 YHI=TABLE(I,XX)                                                   FTW00640
      YLO=TABLE(I-1,XX)                                                 FTW00650
      DY=(YLO-YHI)*(VZ(I)-Z)/(VZ(I)-VZ(I-1))                            FTW00660
      FTWOV=YHI+DY                                                      FTW00670
  100 RETURN                                                            FTW00680
  200 WRITE (6,1) Z,I1,I2                                               FTW00690
      STOP                                                              FTW00700
  210 WRITE (6,2) I1,I2                                                 FTW00710
      STOP                                                              FTW00720
      END                                                               FTW00730
```

```
      FUNCTION PAINT(I,X)
C     PARABOLIC INTERPOLATION BETWEEN POINTS (J-1)AND (J),
C     FOLLOWING THE PARABOLA PASSING THROUGH POINTS (J-1),(J),(J+1).
C     EXCEPT BEFORE LAST POINT.
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT
    1 FORMAT (1H0/1H0,6X,6H** X =,1PE15.6,10X,22HOUTSIDE PAR INT TABLE ,
     1I3,3H **//,37X,25H(PARABOLIC INTERPOLATION))
C     T1=ABSCISSA.    T2=ORDINATE
      K=I
C     K= TABLE INDEX.     J=POINT INDEX.
      NNN=NP(K)
      NN=NP(K)-1
      IF(X-T1(1,K)) 5,20,30
    5 IF(X-T1(1,K)+0.00001*ABS(T1(1,K)))10,10,20
   10 WRITE (6,1)X,K
      STOP
   20 PAINT=T2(1,K)
      GO TO 100
   30 IF(X-T1(NN,K))40,40,31
   31 IF(X-T1(NNN,K))32,33,35
   32 J=NN
      GO TO 60
   33 J=NNN
      PAINT=T2(J,K)
      GO TO 100
   35 IF(X-T1(NNN,K)-0.00001*ABS(T1(NNN,K)))33,10,10
   40 DO 50 J=2,NN
      KJ=J
      IF(X-T1(J,K))60,60,50
   50 CONTINUE
   60 DZ=(T1(J-1,K)-T1(J,K))*(T1(J-1,K)-T1(J+1,K))
      D1=(T1(J,K)-T1(J-1,K))*(T1(J,K)-T1(J+1,K))
      D2=(T1(J+1,K)-T1(J-1,K))*(T1(J+1,K)-T1(J,K))
      CLZ=((X-T1(J,K))*(X-T1(J+1,K)))/DZ
      CL1=((X-T1(J-1,K))*(X-T1(J+1,K)))/D1
      CL2=((X-T1(J-1,K))*(X-T1(J,K)))/D2
      PAINT=T2(J-1,K)*CLZ+T2(J,K)*CL1+T2(J+1,K)*CL2
  100 RETURN
      END
```

```
      FUNCTION TABLE(I,X)                                               TABL0020
C     LINEAR INTERPOLATION BETWEEN POINTS (J-1) AND (J).                TABL0030
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT              TABL0040
    1 FORMAT (1H0/1H0,6X,6H** X =,E15.6,10X,14HOUTSIDE TABLE ,I3,3H **//TABL0050
     1,37X,22H(LINEAR INTERPOLATION))                                   TABL0060
C     T1=ABSCISSA.    T2=ORDINATE                                       TABL0070
      K=I                                                               TABL0080
C     K= TABLE INDEX.      J=POINT INDEX.                               TABL0090
      NN=NP(K)                                                          TABL0100
      IF(X-T1(1,K)) 5,20,30                                             TABL0110
    5 IF(X-T1(1,K)+0.00001*ABS(T1(1,K)))10,10,20                        TABL0120
   10 WRITE (6,1)X,K                                                    TABL0130
      STOP                                                              TABL0140
   20 TABLE=T2(1,K)                                                     TABL0150
      GO TO 100                                                         TABL0160
   30 IF(X-T1(NN,K)) 40,36,35                                           TABL0170
   35 IF(X-T1(NN,K)-0.00001*ABS(T1(NN,K)))36,10,10                      TABL0180
   36 TABLE=T1(NN,K)                                                    TABL0190
      GO TO 100                                                         TABL0200
   40 DO 50 J=2,NN                                                      TABL0210
      KJ=J                                                              TABL0220
      IF(X-T1(J,K)) 60,60,50                                            TABL0230
   50 CONTINUE                                                          TABL0240
   60 TABLE=T2(KJ,K)+(T2(KJ-1,K)-T2(KJ,K))*(T1(KJ,K)-X)/(T1(KJ,K)-T1(KJ-TABL0250
     11,K))                                                             TABL0260
  100 RETURN                                                            TABL0270
      END                                                               TABL0280
```

```
      FUNCTION DELAY (I,TAU,STEP,XDEL)                                  DELA0020
C     FIXED OR VARIABLE DELAY FUNCTION CALLED BY DER.                   DELA0030
C     OUTPUT USES LINEAR INTERPOLATION  BETWEEN POINTS K AND KP         DELA0040
C     T1=ABSCISSA.    T2=ORDINATE                                       DELA0050
C                                                                       DELA0060
C                    TTAU                           T    ATT            DELA0070
C               +STEP-+  +---------------TAU----------+    +            DELA0080
C               /*****/*****/*****/*****/*****/*****/*****/            DELA0090
C                     K      KP                      NP                 DELA0100
C                                                                       DELA0110
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT              DELA0120
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1   DELA0130
  500 FORMAT(1H0,6X,39HOVERLAP BETWEEN TABLE AND DELAY STORAGE/7X,12HNO DELA0140
     1EXECUTION//)                                                      DELA0150
C                                                                       DELA0160
C                                                                       DELA0170
      DATA NSLA/100/                                                    DELA0180
C     NSLA=CAPACITY AVAILABLE FOR EACH DELAY                            DELA0190
C     NSLA   MUST BE CHANGED IF THE DIMENSIONS OF THE COMMON            DELA0200
C     /ELBAT/ ARE CHANGED                                               DELA0210
C                                                                       DELA0220
C                                                                       DELA0230
C     INITIAL CHECK                                                     DELA0240
      NN=I                                                              DELA0250
      IF(I.GT.NT) GO TO 190                                             DELA0260
      WRITE(6,500)                                                      DELA0270
      STOP                                                              DELA0280
  190 IF(STEP-DT)200,300,300                                            DELA0290
  200 DT=STEP                                                           DELA0300
      HMAX=STEP                                                         DELA0310
C                                                                       DELA0320
C                                                                       DELA0330
C     *****************LOADING OF TABLES***************************     DELA0340
C     ATT IS THE NEXT LOADING TIME                                      DELA0350
  300 IF(T.LT.ATT(NN))  GO TO 3                                         DELA0360
      IF(T.GT.1.E-30) GO TO 301                                         DELA0370
      NP(NN)=0                                                          DELA0380
      ATT(NN)=-0.00001*STEP                                             DELA0390
  301 NP(NN)=NP(NN)+1                                                   DELA0400
      IF(NP(NN)-NSLA)11,11,9                                            DELA0000
    9 NP(NN)=1                                                          DELA0410
   11 M=NP(NN)                                                          DELA0420
      T1(M,NN)=T                                                        DELA0430
      T2(M,NN)=XDEL                                                     DELA0440
      ATT(NN)=ATT(NN)+STEP                                              DELA0450
```

```
      3 CONTINUE                                                        DELA0460
C                                                                       DELA0470
C                                                                       DELA0480
C       *****************READING OF TABLES****************************  DELA0490
        IF(T-TAU)4,4,5                                                  DELA0500
C                                                                       DELA0510
C       BEGIN WITH INITIAL VALUE                                        DELA0520
      4 DELAY=T2(1,NN)                                                  DELA0530
        CALL START(TAU)                                                 DELA0540
        IF(NP(NN)-1)14,14,100                                           DELA0550
     14 T1(NSLA,NN)=-STEP                                               DELA0560
        T2(NSLA,NN)=DELAY                                               DELA0570
        GO TO 100                                                       DELA0580
C                                                                       DELA0590
C       TRUE DELAY                                                      DELA0600
      5 CONTINUE                                                        DELA0610
        SSS=ATT(NN)-T                                                   DELA0620
        SSS=STEP-SSS                                                    DELA0630
        K=((TAU-SSS)/STEP)+1.                                           DELA0640
        IF(K-NSLA)55,56,56                                              DELA0650
     56 WRITE (6,57)NN                                                  DELA0660
     57 FORMAT(1H0, 6X,50H**TABLE CAPACITY EXCEEDED BY DELAY FUNCTION NUMBDELA0670
       1ER,I3,2H**)                                                     DELA0680
        STOP                                                            DELA0690
     55 CONTINUE                                                        DELA0700
        K=NP(NN)-K                                                      DELA0710
        IF(K)6,6,7                                                      DELA0720
      6 K=K+NSLA                                                        DELA0730
      7 TTAU=T-TAU                                                      DELA0740
        KP=K+1                                                          DELA0750
        IF(KP-NSLA)8,8,10                                               DELA0760
     10 KP=1                                                            DELA0770
      8 CONTINUE                                                        DELA0780
        DELAY=T2(KP,NN)+(T2(K,NN)-T2(KP,NN))*(T1(KP,NN)-TTAU)/(T1(KP,NN)-TDELA0790
       11(K,NN))                                                        DELA0800
    100 RETURN                                                          DELA0810
        END                                                             DELA0820
```

```
      SUBROUTINE FIT(NTAB,IORD,A)                                       FIT 0020
C     LEAST SQUARES FIT                                                 FIT 0030
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT              FIT 0040
      DIMENSION V(100),S(18),A(10),COMA(10,10),C(10)                    FIT 0050
C   CALCULATES THE COEFFICIENTS A(K) OF THE POLYNOM SUM A(K)*X**(K-1),K=FIT 0060
C     1,N,WHICH BEST FITS IN THE LEAST SQUARE SENSE A GIVEN SET OF M POIFIT 0070
C     NTS (X,Y).                                                        FIT 0080
C                                                                       FIT 0090
      M=NP(NTAB)                                                        FIT 0100
      N=IORD+1                                                          FIT 0110
C   CALCULATION OF THE MATRIX ELEMENTS COMA(K,L)                        FIT 0120
      DO 6 K=1,N                                                        FIT 0130
    6 A(K)=0.                                                           FIT 0140
      DO 5 I=1,M                                                        FIT 0150
    5 V(I)=1.                                                           FIT 0160
      MG=2*(N-1)                                                        FIT 0170
      DO 20 NPK=1,MG                                                    FIT 0180
      S(NPK)=0.                                                         FIT 0190
      IF(NPK.GT.N) GO TO 10                                             FIT 0200
      C(NPK)=0.                                                         FIT 0210
      DO 8 I=1,M                                                        FIT 0220
    8 C(NPK)=C(NPK)+V(I)*T2(I,NTAB)                                     FIT 0230
   10 DO 18 I=1,M                                                       FIT 0240
      V(I)=V(I)*T1(I,NTAB)                                              FIT 0250
   18 S(NPK)=S(NPK)+V(I)                                                FIT 0260
   20 CONTINUE                                                          FIT 0270
      COMA(1,1)=FLOAT(M)                                                FIT 0280
      DO 30 J=2,N                                                       FIT 0290
      COMA(J,J)=S(2*J-2)                                                FIT 0300
      JM1=J-1                                                           FIT 0310
      DO 23 L=1,JM1                                                     FIT 0320
      JPL=J+L-2                                                         FIT 0330
      COMA(J,L)=S(JPL)                                                  FIT 0340
   23 COMA(L,J)=S(JPL)                                                  FIT 0350
   30 CONTINUE                                                          FIT 0360
C     STEPS OF ELIMINATION PROCESS                                      FIT 0370
      N1 = N-1                                                          FIT 0380
      DO 90 J=1,N1                                                      FIT 0390
      J1 = J+1                                                          FIT 0400
      DO 90 K=J1,N                                                      FIT 0000
      Q = COMA(K,J)/COMA(J,J)                                           FIT 0410
      DO 110 L=J1,N                                                     FIT 0420
  110 COMA(K,L) = COMA(K,L)-Q*COMA(J,L)                                 FIT 0430
   90 C(K) = C(K)-Q*C(J)                                                FIT 0440
C     CALCULATION OF THE COEFFICIENTS OF THE POLYNOM                    FIT 0450
```

```
      A(N) = C(N)/COMA(N,N)                                             FIT 0460
      I= N                                                              FIT 0470
  140 P = 0.0                                                           FIT 0480
      DO 120 L=I,N                                                      FIT 0490
  120 P = P+COMA(I-1,L)*A(L)                                            FIT 0500
      I = I-1                                                           FIT 0510
      Q = C(I)-P                                                        FIT 0520
      A(I) = Q/COMA(I,I)                                                FIT 0530
      IF (I-1) 130,130,140                                              FIT 0540
  130 RETURN                                                            FIT 0550
      END                                                               FIT 0560
```

```
      SUBROUTINE LOAD(I,STEP,XSTO)
C     LOADS VARIABLE XSTO IN TABLE I
      COMMON/ELBAT/T1(100,30),T2(100,30),NP(31),ATT(30),NT
      COMMON/VARIAB/NEQ,T,X(150),DX(150),DT,E,F,HMIN,HMAX,BETA,A,IOP1
    1 FORMAT (1H0,6X,10HLOAD TABLE ,I3,35H EXCEEDS CAPACITY EXECUTION ST
     1OPPED //)
    2 FORMAT(1H0,6X,10HLOAD TABLE ,I3,6H POINT ,I3,7H TIME = ,1PE12.5,
     13H Y= ,1PE12.5)
      DATA NSLA/100/
C
      IF(STEP.GT.DT) GO TO 10
      DT=STEP
      HMAX=STEP
C
   10 CONTINUE
C     ATT IS THE NEXT LOADING TIME
      IF(T.LT.ATT(I)) GO TO 100
      IF(T.GT.1.E-30) GO TO 20
      NP(I)=0
      ATT(I)=-0.00001*STEP
   20 NP(I)=NP(I)+1
      IF(NP(I).GT.NSLA) GO TO 110
      M=NP(I)
      T1(M,I)=T
      T2(M,I)=XSTO
      WRITE(6,2) I,M,T1( M,I),T2( M,I)
      ATT(I)=ATT(I)+STEP
  100 RETURN
  110 WRITE (6,1) I
      STOP
      END
```

```
      SUBROUTINE REPEAT(X,XZ,PAR)
C     DUMMY REPEAT
      CALL STOPR
      RETURN
      END
```

```
PTII     TITLE  SUBROUTINE COEFF
COEFF    START 0
         ENTRY WRITE
         ENTRY WRPP
         ENTRY DRAW
         ENTRY DRAW1
         ENTRY OUTPUT
         ENTRY DRAWXY
         ENTRY DRAW2
         EXTRN UTIL,STAMPA
         EXTRN CALCOM
         EXTRN VARIAB
         SPACE 2
         BC    15,12 0,15
         DC    X 070000
         DC    C COEFF
         SAVE   14,12
         BALR  12,0
         USING *,12
         ST    13,SAREA 4
         LA    13,SAREA
         EXTRN STAMP1
         SR    2,2
         SR    5,5
COUNT    L     6,0 5,1
         LTR   6,6
         BM    NEG2
         LE    0,0 6
         STE   0,VN1 2
         LA    5,4 5
         L     6,0 5,1
         LTR   6,6
         BM    NEG1
         LE    0,0 6
         STE   0,VE1 2
         LA    5,4 5
         LA    2,4 2
         C     2,TWOH
         BNH   COUNT
NEG1     LE    0,0 6
         STE   0,VE1 2
NEG2     LA    2,4 2
         SRA   2,2
         ST    2,NCOE
         CALL  STAMP1, VN1,VE1,NCOE
         L     13,SAREA 4
         RETURN  14,12
VE1      DS    51F
VN1      DS    51F
NCOE     DC    F 0
TWOH     DC    F 200
SAREA    DS    18F
         TITLE  SUBROUTINE OUTPUT
OUTPUT   BC    15,12 0,15
         DC    X 0700
         DC    C OUTPUT
         SAVE   14,12
         BALR  12,0
         USING *,12
         USING COMMON,7
         L     7,ACOMM
```

```
         SR    2,2                                                      PUT00610
CONTO    L     6,0 2,1                                                  PUT00620
         LTR   6,6                                                      PUT00630
         BM    NEG3                                                     PUT00640
         LE    0,0 6                                                    PUT00650
         STE   0,NS 2                                                   PUT00660
         LA    2,4 2                                                    PUT00670
         C     2,C148                                                   PUT00680
         BNH   CONTO                                                    PUT00690
NEG3     LE    0,0 6                                                    PUT00700
         STE   0,NS 2                                                   PUT00710
         LA    2,4 2                                                    PUT00720
         SRA   2,2                                                      PUT00730
         ST    2,N1                                                     PUT00740
         RETURN  14,12                                                  PUT00750
C148     DC    F 592                                                    PUT00760
         TITLE  DRAW                                                    PUT00770
DRAW     BC    15,12 0,15                                               PUT00780
         DC    X 07000000                                               PUT00790
         DC    C DRAW                                                   PUT00800
         STM   14,12,12 13                                              PUT00810
         BALR  12,0                                                     PUT00820
         USING *,12                                                     PUT00830
         USING COMCAL,11,7,10                                           PUT00840
         L     11,ACAL                                                  PUT00850
         L     10, F 4096                                               PUT00860
         LA    7,0 11,10                                                PUT00870
         LA    10,0 7,10                                                PUT00880
         L     2,NCALL                                                  PUT00890
         LA    2,1 2                                                    PUT00900
         ST    2,NCALL                    NCALL NCALL 1  REG.2          PUT00910
         SLA   2,2                                                      PUT00920
         ST    1,ACALL-4 2                                              PUT00930
*                        COUNT OF PARAMETERS                            PUT00940
         SR    4,4                                                      PUT00950
         ST    4,NOCALC                                                 PUT00960
LOOP5    L     6,0 4,1                                                  PUT00970
         LTR   6,6                                                      PUT00980
         BM    NEG4                                                     PUT00990
         LA    4,4 4                                                    PUT01000
         B     LOOP5                                                    PUT01010
NEG4     LA    4,4 4                                                    PUT01020
         SRA   4,3                        NO.PAR REG4/8                 PUT01030
         ST    4,NCURV-4 2                                              PUT01040
*                                         REG.4   NO. PAR.              PUT01050
         L     5,NA                       REG.5   PREVIOUS NA           PUT01060
         LA    6,0 5,4                                                  PUT01070
         SLA   5,2                                                      PUT01080
         ST    6,NAA                      REG.6   NEW NA                PUT01090
         C     6,FIFTEEN                    COMPARE NEW NA              PUT01100
         BNH   NORM                         WITH 15                     PUT01110
         B     FINERR                                                   PUT01120
NORM     LA    8,HCURV 5                                                PUT01130
         SR    9,9                                                      PUT01140
         SR    3,3                                                      PUT01150
LOOP6    L     11,0 3,1                                                 PUT01160
         LE    0,0 11                                                   PUT01170
         STE   0,0 8,9                                                  PUT01180
         LA    9,4 9                                                    PUT01190
         LA    3,8 3                                                    PUT01200
         BCT   4,LOOP6                                                  PUT01210
```

```
         ST     6,NA
FINRIG   RETURN  14,12
FINERR   BCT    2,FIN1
FIN1     ST     2,NCALL
         B      FINRIG
         TITLE   DRAW1
DRAW1    BC     15,12 0,15
         DC     X 070000
         DC     C DRAW1
         STM    14,12,12 13
         BALR   12,0
         USING  *,12
         USING  COMCAL,11,7,10
         L      11,ACAL
         L      10, F 4096
         LA     7,0 11,10
         LA     10,0 7,10
         L      6,0 1
         L      0,0 1
         SR     5,5
         SR     9,9
E1       L      2,ACALL 9
         SR     4,4
E2       LA     3,0 4
         SLA    3,3                            *8
         LA     8,4 2,3
         L      3,0 8
         LE     0,0 3
         STE    0,0 6
         LA     6,512 6
         LA     4,1 4
         C      4,NCURV 9
         BL     E2
         LA     5,1 5
         LA     9,4 9
         C      5,NCALL
         BL     E1
         USING  IABVAR,4
         L      4,AVAR
         LE     0,T
         A      0, F 7680
         LR     3,0
         STE    0,0 3
         RETURN  14,12
         TITLE   DRAWXY
DRAWXY   BC     15,12 0,15
         DC     X 070C
         DC     C DRAWXY
         STM    14,12,12 13
         BALR   12,0
         USING  *,12
         USING  COMCAL,11,7,10
         L      11,ACAL
         L      10, F 4096
         LA     7,0 11,10
         LA     10,0 7,10
         L      2,NCALLX
         LA     2,1 2
         ST     2,NCALLX
         SLA    2,2                 NCALLX NCALLX 1 REG 2
         ST     1,ACALLX-4 2
```

```
*
         SR     4,4                                   COUNT OF PARAMETERS
         ST     4,NOXY
LOOP7    L      6,0 4,1
         LTR    6,6
         BM     NEG6
         LA     4,4 4
         B      LOOP7
NEG6     LA     4,4 4
         SRA    4,3                          NO.PAR REG4/8   REG.4
         ST     4,NCURVX-4 2
*
         L      5,NAXY
         LA     6,0 5,4
         SLA    5,2
         C      6,SIX
         BNH    NORMX
         B      FINERRX
NORMX    LA     8,HCURVX 5
         SR     9,9
         SR     3,3
LOOP8    L      11,0 3,1
         LE     0,0 11
         STE    0,0 8,9
         LA     9,4 9
         LA     3,8 3
         BCT    4,LOOP8
         ST     6,NAXY
FINRIGX  RETURN  14,12
FINERRX  BCT    2,FIN1X
FIN1X    ST     2,NCALLX
         B      FINRIGX
         TITLE   DRAW2
DRAW2    BC     15,12 0,15
         DC     X 070000
         DC     C DRAW2
         STM    14,12,12 13
         BALR   12,0
         USING  *,12
         USING  COMCAL,11,7,10
         L      11,ACAL
         L      10, F 4096
         LA     7,0 11,10
         LA     10,0 7,10
         L      6,0 1
         SR     5,5
         SR     9,9
E1X      L      2,ACALLX 9
         SR     4,4
E2X      LA     3,0 4
         SLA    3,3                                   *8
         LA     8,4 2,3
         L      3,0 8
         LE     0,0 3
         STE    0,0 6
         LA     6,512 6
         LA     4,1 4
         C      4,NCURVX 9
         BL     E2X
         LA     5,1 5
         LA     9,4 9
```

```
         C       5,NCALLX
         BL      E1X
         RETURN   14,12
         TITLE    WRITE
WRITE    BC      15,12 0,15
         DC      X 070000
         DC      C WRITE
         STM     14,12,12 13
         BALR    12,0
         USING   *,12
         LA      11,INIZ1
         B       WRPP1
         TITLE    WRPP
WRPP     BC      15,12 0,15
         DC      X 07000000
         DC      C WRPP
         STM     14,12,12 13
         BALR    12,0
         USING   *,12
         LA      11,INIZ2
WRPP1    BALR    12,0
         USING   *,12
         USING   COMMON,7
         LR      4,13
         LA      13,AREA1
         ST      13,8 0,4
         ST      4,4 0,13
         L       7,ACOMM
         BR      11
         SPACE   2
INIZ1    SR      2,2
         ST      1,ORGP
LOOP1    LE      0,0 2,1
         LTER    0,0
         BM      NEG
         LA      2,4 2
         B       LOOP1
NEG      LA      2,4 2
         SRA     2,3
         C       2,FIFTY
         BNH     STORE
         L       2,FIFTY
STORE    ST      2,N2
         SR      5,5
         SR      6,6
LOOP2    L       11,0 5,1
         LE      0,0 11
         STE     0,VN 6
         LA      5,8 5
         LA      6,4 6
         BCT     2,LOOP2
         SPACE   2
RET      L       13,AREA1 4
         LM      2,12,28 13
         L       14,12 13
         MVI     12 13 ,X FF
         BR      14
         SPACE   2
INIZ2    L       2,N2
         LTR     2,2
         BZ      CALLST
```

```
         L     1,ORGP                                                   PUT03050
         SR    5,5                                                      PUT03060
         SR    6,6                                                      PUT03070
LOOP3    L     11,4 5,1                                                 PUT03080
         LE    0,0 11                                                   PUT03090
         STE   0,VE 6                                                   PUT03100
         LA    5,8 5                                                    PUT03110
         LA    6,4 6                                                    PUT03120
         BCT   2,LOOP3                                                  PUT03130
CALLST   L     15,ASTAMP                                                PUT03140
         BALR  14,15                                                    PUT03150
         B     RET                                                      PUT03160
AREA1    DS    18F                                                      PUT03170
ORGP     DS    1F                                                       PUT03180
         SPACE 2                                                        PUT03190
AVAR     DC    A VARIAB                                                 PUT03200
ACAL     DC    A CALCOM                                                 PUT03210
ACOMM    DC    A UTIL                                                   PUT03220
ASTAMP   DC    A STAMPA                                                 PUT03230
ACALL    DS    16F                                                      PUT03240
ACALLX   DS    16F                                                      PUT03250
I2       DC    F 0                                                      PUT03260
SIX      DC    F 6                                                      PUT03270
FIFTEEN  DC    F 15                                                     PUT03280
FIFTY    DC    F 50                                                     PUT03290
NAA      DS    1F                                                       PUT03300
         EJECT                                                          PUT03310
*                                                                       PUT03320
COMCAL   DSECT                                                          PUT03330
TABTX    DS    2048F                         C                          PUT03340
HCURV    DS    15F                           O                          PUT03350
NCALL    DS    1F                            M                          PUT03360
NCURV    DS    15F                           M                          PUT03370
TABXY    DS    768F                          O                          PUT03380
HCURVX   DS    6F                            N                          PUT03390
NCALLX   DS    1F                                                       PUT03400
NCURVX   DS    6F                                                       PUT03410
KA       DS    1F                            C                          PUT03420
NA       DS    1F                            A                          PUT03430
NAXY     DS    1F                            L                          PUT03440
NOCALC   DS    1F                            C                          PUT03450
HEAD     DS    18F                           O                          PUT03460
NOXY     DS    1F                            M                          PUT03470
*                                                                       PUT03480
COMMON   DSECT                                                          PUT03490
NS       DS    150F                                                     PUT03500
N1       DS    1F                                                       PUT03510
N2       DS    1F                                                       PUT03520
N3       DS    1F                                                       PUT03530
ID       DS    1F                                                       PUT03540
VN       DS    50F                                                      PUT03550
VE       DS    50F                                                      PUT03560
CXZ1     DS    150F                                                     PUT03570
CXZ2     DS    150F                                                     PUT03580
CXZ3     DS    150F                                                     PUT03590
CXZ4     DS    150F                                                     PUT03600
CXZ5     DS    150F                                                     PUT03610
CXZ6     DS    150F                                                     PUT03620
CXZ7     DS    150F                                                     PUT03630
CXZ8     DS    150F                                                     PUT03640
CXZ9     DS    150F                                                     PUT03650
```

```
CXZA     DS      150F                                            PUT03660
*                                                                PUT03670
IABVAR   DSECT                                                   PUT03680
NEQ      DS      1F                                              PUT03690
T        DS      1F                                              PUT03700
X        DS      150F                                            PUT03710
DX       DS      150F                                            PUT03720
DT       DS      1F                                              PUT03730
E        DS      1F                                              PUT03740
F        DS      1F                                              PUT03750
HMIN     DS      1F                                              PUT03760
HMAX     DS      1F                                              PUT03770
BETA     DS      1F                                              PUT03780
A        DS      1F                                              PUT03790
IOP1     DS      1F                                              PUT03800
         END                                                     PUT03810
```

Appendix II

Programme for the IBM 7090/94.

All the programme is identical with the programme for the 360/65, with the following exception:

1) In the labeled common ELBAT the dimensions are reduced to T 1(100,10), T2(100,10) for storage limitations.

2) The subroutine PUTØUT, which is written in ASSEMBLER language for the 360, must be replaced by the subroutine in MAP given in the following pages. These subroutines handle output statements.

```
$IBMAP PUTOUT  200                                                PUT00000
       USE     DHO                                                PUT00010
       ENTRY   WRITE                                              PUT00020
       ENTRY   WRPP                                               PUT00030
       ENTRY   COEF                                               PUT00040
       ENTRY   DRAW                                               PUT00050
       ENTRY   DRAW1                                              PUT00060
       ENTRY   DRAWXY                                             PUT00070
       ENTRY   DRAW2                                              PUT00080
       REM                   CALL WRITE (1HA,A,4HBETA,B)          PUT00090
 WRITE SAVE    (1,4)                                              PUT00100
       SXA     COM,4                                              PUT00110
       CLA     1,4                                                PUT00120
       ANA     =077000000                                         PUT00130
       ARS     19                                                 PUT00140
       STO     N2                                                 PUT00150
       PAC     0,1                                                PUT00160
       SXD     R8,1                                               PUT00170
       SXD     R4,1                                               PUT00180
       AXT     0,1                                                PUT00190
    R3 CLA*    3,4                                                PUT00200
       STO     VN,1                                               PUT00210
       TXI     *+1,4,-2                                           PUT00220
       TXI     *+1,1,-1                                           PUT00230
    R4 TXH     R3,1,**       -N2=-NUMBER ARG /2.                  PUT00240
       RETURN  WRITE                                              PUT00250
       REM                   CALL WRPP                            PUT00260
  WRPP SAVE    (1,4)                                              PUT00270
       CLA     N2                                                 PUT00280
       TZE     R8+1                                               PUT00290
   COM AXT     **,4                                               PUT00300
       AXT     0,1                                                PUT00310
    R7 CLA*     ,4                                                PUT00320
       STO     VE,1                                               PUT00330
       TXI     *+1,4,-2                                           PUT00340
       TXI     *+1,1,-1                                           PUT00350
    R8 TXH     R7,1,**       -N2=-NUMBER ARG WRITE/2              PUT00360
       CALL    STAMPA                                             PUT00370
       RETURN  WRPP                                               PUT00380
       REM                   CALL COEF(1HB,BETA,4HALFA,A)         PUT00390
  COEF SAVE    (1,4)                                              PUT00400
       CLA     1,4                                                PUT00410
       ANA     =077000000                                         PUT00420
       ARS     19                                                 PUT00430
       STO     NCOE                                               PUT00440
```

```
      PAC     0,1                                            PUT00450
      SXD     R6,1                                           PUT00460
      AXT     0,1                                            PUT00470
R5    CLA*    3,4                                            PUT00480
      STO     VN1,1                                          PUT00490
      CLA*    4,4                                            PUT00500
      STO     VE1,1                                          PUT00510
      TXI     *+1,4,-2                                       PUT00520
      TXI     *+1,1,-1                                       PUT00530
R6    TXH     R5,1,**          -NCOE                         PUT00540
      CALL    STAMP1(VN1,VE1,NCOE)                           PUT00550
      RETURN  COEF                                           PUT00560
      REM                      CALL DRAW(1HA,A,1HB,B)        PUT00570
NC    PZE     0                                              PUT00580
DRAW  SAVE    (1,2,4)                                        PUT00590
      CLA     NCALL                                          PUT00600
      ADD     =1                                             PUT00610
      STO     NCALL                                          PUT00620
      PAC     0,1                                            PUT00630
      PXA     0,4                                            PUT00640
      STO     ACALL-1,1                                      PUT00650
      CLA     1,4                                            PUT00660
      ANA     =036000000                                     PUT00670
      ARS     19                                             PUT00680
      PAC     0,2              - NUMBER OF CURVES            PUT00690
      SXD     R12,2                                          PUT00700
      STO     NCURV-1,1                                      PUT00710
      ALS     7                                              PUT00720
      PAC     0,2                                            PUT00730
      PXD     0,2                                            PUT00740
      STO     DECR-1,1                                       PUT00750
      CLA     NA                                             PUT00760
      ADD     NCURV-1,1                                      PUT00770
      SUB     =15                                            PUT00780
      TZE     *+2                                            PUT00790
      TPL     RERR                                           PUT00800
      CLA     AHCUR                                          PUT00810
      ADD     NA                                             PUT00820
      STA     R11+1                                          PUT00830
      AXT     0,2                                            PUT00840
R11   CLA*    3,4                                            PUT00850
      STO     **,2             HCURV+NA PREVIOUS             PUT00860
      TXI     *+1,4,-2                                       PUT00870
      TXI     *+1,2,-1                                       PUT00880
```

```
R12     TXH     R11,2,**        - NCURV(NCALL)                  PUT00890
        CLA     NA                                              PUT00900
        ADD     NCURV-1,1                                       PUT00910
        STO     NA                                              PUT00920
        STZ     NOCALC                                          PUT00930
FINE    RETURN  DRAW                                            PUT00940
RERR    CLA     NCALL                                           PUT00950
        SUB     =1                                              PUT00960
        STO     NCALL                                           PUT00970
        TRA     FINE                                            PUT00980
AHCUR   PZE     HCURV                                           PUT00990
DRAWXY  SAVE    (1,2,4)         CALL DRAWXY(1HA,A,1HB,B)        PUT01000
        CLA     NCALLX                                          PUT01010
        ADD     =1                                              PUT01020
        STO     NCALLX                                          PUT01030
        PAC     0,1             XR1=-NCALLX                     PUT01040
        PXA     0,4                                             PUT01050
        STO     ACALLX-1,1                                      PUT01060
        CLA     1,4                                             PUT01070
        ANA     =017000000                                      PUT01080
        ARS     19                                              PUT01090
        PAC     0,2             -NUMBER OF CURVES               PUT01100
        SXD     S12,2                                           PUT01110
        STO     NCURVX-1,1                                      PUT01120
        ALS     7                                               PUT01130
        PAC     0,2                                             PUT01140
        PXD     0,2                                             PUT01150
        STO     DECRX-1,1                                       PUT01160
        CLA     NAXY                                            PUT01170
        ADD     NCURVX-1,1                                      PUT01180
        SUB     =6                                              PUT01190
        TZE     *+2                                             PUT01200
        TPL     SERR                                            PUT01210
        CLA     AHCURX                                          PUT01220
        ADD     NAXY                                            PUT01230
        STA     S11+1                                           PUT01240
        AXT     0,2                                             PUT01250
S11     CLA*    3,4                                             PUT01260
        STO     **,2            HCURVX+NAXY PREVIOUS            PUT01270
        TXI     *+1,4,-2                                        PUT01280
        TXI     *+1,2,-1                                        PUT01290
S12     TXH     S11,2,**        -NCURVX(NCALLX)                 PUT01300
        CLA     NAXY                                            PUT01310
        ADD     NCURVX-1,1                                      PUT01320
        STO     NAXY                                            PUT01330
```

```
         STZ    NOXY                                               PUT01340
   FINEX RETURN DRAWXY                                             PUT01350
   SERR  CLA    NCALLX                                             PUT01360
         SUB    =1                                                 PUT01370
         STO    NCALLX                                             PUT01380
         TRA    FINEX                                              PUT01390
  AHCURX PZE    HCURVX                                             PUT01400
     LAV PZE    0                                                  PUT01410
         REM                    CALL DRAW1(TABTX)                  PUT01420
   DRAW1 SAVE   (1,2,4)                                            PUT01430
         STZ    I2                                                 PUT01440
         CLA    NCALL                                              PUT01450
         PAC    0,1                                                PUT01460
         SXD    R22,1                                              PUT01470
         AXT    0,2                                                PUT01480
     R18 CLA    I2                                                 PUT01490
         ALS    7                                                  PUT01500
         STO    LAV                                                PUT01510
         CLA    3,4                                                PUT01520
         ADD    LAV                                                PUT01530
         STA    R20                                                PUT01540
         CLA    DECR,2                                             PUT01550
         STD    R21                                                PUT01560
         CLA    ACALL,2                                            PUT01570
         PAX    0,1                                                PUT01580
         SXA    REST,2          SAVE XR2                           PUT01590
         AXT    0,2                                                PUT01600
     R19 CLA*   4,1                                                PUT01610
     R20 STO    **,2                                               PUT01620
         TXI    *+1,1,-2                                           PUT01630
         TXI    *+1,2,-128                                         PUT01640
     R21 TXH    R19,2,**                                           PUT01650
    REST AXT    **,2            RESTORE XR2                        PUT01660
         CLA    NCURV,2                                            PUT01670
         ADD    I2                                                 PUT01680
         STO    I2                                                 PUT01690
         TXI    *+1,2,-1                                           PUT01700
     R22 TXH    R18,2,**        - NCALL                            PUT01710
         CLA    3,4                                                PUT01720
         ADD    =1920                                              PUT01730
         STA    TT                                                 PUT01740
         CLA    T                                                  PUT01750
      TT STO    **                                                 PUT01760
         RETURN DRAW1                                              PUT01770
```

```
 DRAW2 SAVE    (1,2,4)
       STZ     I2
       CLA     NCALLX
       PAC     0,1
       SXD     S22,1
       AXT     0,2
   S18 CLA     I2
       ALS     7
       STO     LAV
       CLA     3,4
       ADD     LAV
       STA     S20
       CLA     DECRX,2
       STD     S21
       CLA     ACALLX,2
       PAX     0,1
       SXA     SEST,2          SAVE XR2
       AXT     0,2
   S19 CLA*    4,1
   S20 STO     **,2
       TXI     *+1,1,-2
       TXI     *+1,2,-128
   S21 TXH     S19,2,**
  SEST AXT     **,2            RESTORE XR2
       CLA     NCURVX,2
       ADD     I2
       STO     I2
       TXI     *+1,2,-1
   S22 TXH     S18,2,**        -N CALLX
       RETURN  DRAW2
    I2 PZE     0
 ACALL BSS     16
  DECR BSS     16
ACALLX BSS     6
 DECRX BSS     6
   VN1 BSS     50
   VE1 BSS     50
  NCOE PZE     0
       REM                     CALL OUTPUT(1,5,9)
       ENTRY   OUTPUT
OUTPUT SAVE    (1,4)
       CLA     1,4
       ANA     =077000000
       ARS     18
       STO     N1
```

```
         PAC     0,1                                          PUT02230
         SXD     R2,1                                         PUT02240
         AXT     0,1                                          PUT02250
R1       CLA*    3,4                                          PUT02260
         STO     NS,1                                         PUT02270
         TXI     *+1,4,-1                                     PUT02280
         TXI     *+1,1,-1                                     PUT02290
R2       TXH     R1,1,**          -(NA)                       PUT02300
         RETURN  OUTPUT                                       PUT02310
CALCOM   CONTRL  CALCOM                                       PUT02320
         USE     CALCOM                                       PUT02330
         EVEN                                                 PUT02340
TABTX    BSS     2048                                         PUT02350
HCURV    BSS     15                                           PUT02360
NCALL    BSS     1                                            PUT02370
NCURV    BSS     15                                           PUT02380
TABXY    BSS     768                                          PUT02390
HCURVX   BSS     6                                            PUT02400
NCALLX   BSS     1                                            PUT02410
NCURVX   BSS     6                                            PUT02420
KA       BSS     1                                            PUT02430
NA       BSS     1                                            PUT02440
NAXY     BSS     1                                            PUT02450
NOCALC   BSS     1                                            PUT02460
HEAD     BSS     18                                           PUT02470
NOXY     BSS     1                                            PUT02480
VARIAB   CONTRL  VARIAB                                       PUT02490
         USE     VARIAB                                       PUT02500
         EVEN                                                 PUT02510
NEQ      BSS     1                                            PUT02520
T        BSS     1                                            PUT02530
X        BSS     150                                          PUT02540
DX       BSS     150                                          PUT02550
DT       BSS     1                                            PUT02551
E        BSS     1                                            PUT02552
F        BSS     1                                            PUT02553
HMIN     BSS     1                                            PUT02554
HMAX     BSS     1                                            PUT02555
BETA     BSS     1                                            PUT02556
A        BSS     1                                            PUT02557
IOP1     BSS     1                                            PUT02558
UTIL     CONTRL  UTIL                                         PUT02560
         USE     UTIL                                         PUT02570
         EVEN                                                 PUT02580
```

```
NS   BSS   150                                             PUT02590
N1   BSS   1                                               PUT02600
N2   BSS   1                                               PUT02610
N3   BSS   1                                               PUT02620
ID   BSS   1                                               PUT02630
VN   BSS   50                                              PUT02640
VE   BSS   50                                              PUT02650
CXZ  BSS   1500                                            PUT02660
     USE   DHO                                             PUT02670
     END                                                   PUT02680
```

NOTICE TO THE READER

All Euratom reports are announced, as and when they are issued, in the monthly periodical **EURATOM INFORMATION**, edited by the Centre for Information and Documentation (CID). For subscription (1 year : US$ 15, £ 5.7) or free specimen copies please write to :

Handelsblatt GmbH
"Euratom Information"
Postfach 1102
D-4 Düsseldorf (Germany)

or

Office central de vente des publications
des Communautés européennes
2, Place de Metz
Luxembourg

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel

SALES OFFICES

All Euratom reports are on sale at the offices listed below, at the prices given on the back of the front cover (when ordering, specify clearly the EUR number and the title of the report, which are shown on the front cover).

OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
2, place de Metz, Luxembourg (Compte chèque postal Nº 191-90)

BELGIQUE — BELGIË
MONITEUR BELGE
40-42, rue de Louvain - Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42, - Brussel

DEUTSCHLAND
BUNDESANZEIGER
Postfach - Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
26, rue Desaix - Paris 15e

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - Roma

LUXEMBOURG
OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
9, rue Goethe - Luxembourg

NEDERLAND
STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

UNITED KINGDOM
H. M. STATIONERY OFFICE
P. O. Box 569 - London S.E.1

EURATOM — C.I.D.
51-53, rue Belliard
Bruxelles (Belgique)