Mercury Library Programme R4440: Solution of Ordinary Differential Equations

Author: Dr A. (Tony) Gibbons

Date: March 1961

This is a rather ingenious program which accepts equations input in the a Mercury Autocode like format, solves them and tabulates the solution.

A scan of the typed write-up follows. I also have Tony's original hand written version.

TITLE A program for the automatic solution of ordinary differential equations.

SUMMARY This program is a complete input routine which accepts other programs describing the differential equations, the initial conditions, and the output required. These items of information are written in a language similar to that of Mercury Autocode.

The equations, which may be of any order up to 25, are solved by the method of power series. The number of equations depends upon the number of terms in each series.

The programs are short, and easy to prepare, and the series method of solution is most efficient.

Although this program uses the same basic method as programs R4440.1 and R4440.3, several new facilities and a more convenient notation have been introduced.

CODING CONVENTIONS

1. The Equations

The independent variable is x and the other variables are denoted by y1, y2, ... The constants π, k1, k2, ... k25 may also be used. There are also the program parameters p1, ... p5.

A variable yi may be defined by means of a differential equation of order I, with the highest derivative on the left hand side. e.g.

$$y1''' = y1\ y1 - 2\ y1\ y1'' - 1 \qquad (1)$$

In this case I initial conditions must be set, viz. y1, y1', and y1'', and the series solution takes the form:

$$a_0 + a_1 x + \dots + a_{I-1} x^{I-1} + \dots + a_s x^s + \dots + a_{s+I} x^{s+I}.$$

i.e. a further s+1 terms of the series are calculated. The number s is normally set to 15, but may be reset.

In a differential equation the right hand side must be a sum of terms. Each term is a constant, or a constant multiple of either one, or two, variables. x is considered to be a variable. Numerical coefficients are written in fixed point form. e.g.

$$y1''' = 4.7\ y1\ y2 - 3.9\ k1\ k2\ k3\ x + 3.7\ y2 - 3\ y4\ y4 + \pi \qquad (2)$$

The variables yi may also be used as auxiliary variables and in this case may be defined either by an equation of the same form as equation (2) e.g.

$$y1 = y2 + y3'\ y4 + 1$$

or as an elementary function of another variable. The functions reciprocal, exponential, logarithm, square root, sine and cosine are available. e.g.

$$y1 = \exp(-4k1\ x) \qquad (3)$$

The argument may only be a constant multiple of a variable.

The function division is also available and in this case two arguments are required:

$$y1 = \text{divide} \; (k1 \; y2, \; 6y3 \;).$$

In the case of sine and cosine both functions are calculated together. Consequently the convention is introduced that

$$y2 = \sin \; (\; y1 \;) \text{ implies } y3 = \cos \; (y1)$$

and

$$y2 = \cos \; (\pi \, x) \text{ implies } y1 = \sin \; (\pi \, x).$$

The words reciprocal, divide, exp, log and cosine may be abbreviated to the first letter, sqrt and sine to the first two letters. The function words may be preceded by a $\curlyvee$ as in Autocode.

The total number of variables y1 - yN must be such that

$$(s + 1) \; (N + 1) + \sum I_i \leqslant 416 \qquad (4)$$

→ where $\sum I_i$ is the sum of the orders of the differential equations.

Equations may be of any length but are terminated by C.R. It is unlikely that the equations will be too lengthy to fit in the machine.

Each variable yi must be defined by means of a single equation. It is not possible to write for example

$$y_1 = \sin \; (x)$$
$$y_1' = 3y_1.$$

2. Output

The value of any variable, any derivative of a variable, or any constant may be printed in fixed or floating point form at each tabulation point. Thus

print

x = 1, 2   y1 = 7

would print x fixed point, with one figure before the decimal point and two figures after, and y1 floating point to seven significant figures. The layout of the output is the same as that of the print instructions.

Print instructions may also be enclosed in brackets, the significance of this is explained below.

3. Initial Conditions

A number of tabulation points, at which results are to be printed, must be specified. (The tabulation interval does not necessarily correspond to the integration "step").

Two conventions are used. Firstly

x = a (b) c (d) e ...

means that results are to be punched at **x** = a, a+b, a+2b, ... c, c+d, . e, ... and secondly x = a/b/c/d/e ...

means that results are to be punched at x = a, b,c,d,e, ...
The two forms can be used together. e.g.

x = 0.1375/0.2 (0.1) 1 (1) 10 (5)

This would be useful when the initial conditions were known at a non-tabular point and the solution was particularly interesting for $x \leqslant 1$.

A range of values such as (5) may also be specified for each condition. The numbers in the initial conditions may be punched in either fixed or floating point form and are terminated by (,) or / within a range of values and by C.R. L.F. or SP. SP. at the end of a range. e.g.

```
yl = 100 (100) 1, 3     yl' = 1 (1) 10
kl = 1/3
(kl) (yl, yl')
```

This means that the equations are first to be solved with

```
           yl = 100,  yl' = 1  and kl = 1
then with  yl = 100   yl' = 1  and kl = 3
then with  yl = 200   yl' = 2  and kl = 1   etc.
```

The directive (kl) (yl, yl') terminates the initial conditions and indicates how, and in what order, the values are to be changed. When two variables are bracketed together e.g.

(yl, yl')

they must assume the same number of values.

When <u>print</u> instructions are enclosed in brackets e.g.

(yl = 4) (yl' = 4)

they are only obeyed when one of the items to be printed has been changed i.e. at the head of a set of results. Whereas an ordinary print instruction merely causes a number to be printed, a bracketed instruction causes an equation to be printed. e.g.

yl = 2.000, + 2 yl' = 2.000, + 0.

Consequently this facility is useful for heading sets of results.

PREPARATION OF A COMPLETE PROGRAM

The way in which the various parts of the program are fitted together is best illustrated by means of an example:

Write a program to integrate $y'' = k \cos(\pi x) \sin(\pi y)$.

```
title
solution of y'' = k cosπx sinπ y
s → 30
yl → 2
equations
y3 = cos (πx)
y4 = sin (πyl)
yl'' = kl y3 y4
print
(yl = 1, 1) (yl' = 1, 1)
x = 1, 1  yl = 6  yl' = 6
range
x = 0 (1) 6
yl = 0 (0.2) 1   yl' = 0 (0.2) 1
kl = 10
(yl) (yl')
```

NOTES

1. The words title, equation, print and range may be abbreviated, only the first letter being relevant.
2. The title may be omitted. It is terminated by two figure shifts.
3. The directive s → 30 fixes the maximum number of terms of the series that can be calculated. When there are less than about 20 equations this can be omitted c.f. relation (4).
4. The directive yl → 2 is a statement of the order of the differential equation.
5. These ranges of values for the initial conditions would produce 36 sets of results.

MACHINE OPERATION

1. Reading in the program

→ The program is a binary one and occupies S2 and S128 - 229. It should be read into the machine with key 2 up and will come to a hoot stop at the end. (n.b. sector 2 must be de-isolated).

A "program" is read by setting H.S. = 0 and pressing I.T.B. The machine, after interpreting the equations, proceeds with the calculation, counting the terms of the series in B4. At the end of the calculation the machine hoots. A new set of initial conditions e.g.

range
x = 0 (1) 2
yl = 4 yl' = 1 kl = 10
(kl)

may be read by setting key 0 and pressing I.T.B. The directive (kl) is a dummy but must be included to terminate the tape.

2. Faults

Faults encountered while reading the tape or while calculating the series cause the machine to hoot, a fault number being displayed in B7.

A list of the faults which are recognised is given at the end of the specification. Those marked by an asterisk are discovered during the calculation, the others during input.

In the case of fault 32 the incorrect instruction is punched out as far as the spurious character.

3. Restart Facilities

The calculation may not be completed for a number of reasons.

1. A parity failure, i.e. the machine has made a mistake. It is usually possible to re-integrate the equations with the same set of initial conditions by setting key 8 and pressing I.T.B.

2. A fault which is encountered during the calculation. All that can be done in this event is to set key 7 and press I.T.B., the print instructions will then be obeyed once.

3. Provision has been made for a calculation to be broken off and continued later. If key 3 is set and I.T.B. pressed then a binary restart tape is punched out.

This may be used as follows:

1. Put the original program in the tape reader.
2. Set H.S. = 0 and press I.T.B.
3. Put key 2 up immediately.

4. The machine will now hoot when it has read the tape.
5. Insert the restart tape in the reader.
6. Press I.T.B. (with key 2 still up). The calculation will now be resumed.
7. Put key 2 down before reading another program.

This technique of using key 2 to stop the machine after it has read the program tape may also be used prior to reading in a new set of initial conditions.

METHOD OF INTEGRATION

When the initial conditions have been set, standard recurrence relations are used to calculate more terms of the power series. For a variable defined by an equation of order I a series is calculated of the form

$$\underbrace{a_0 + a_1 x \ldots a_{I-1} x^{I-1}}_{\text{I initial conditions}} + \ldots + a_s x^s + \ldots a_{s+I} x^{s+I}.$$

A practical radius of convergence is now determined, i.e. a number R for which

$$\sum_{n=s+1-k}^{s+I} \left| a_n R^n \right| \leqslant p1 \left(\left| \sum_{n=0}^{s+I} a_n R^n \right| + p3 \right)$$

and

$$\sum_{n=0}^{s+I} \left| a_n R^n \right| \leqslant p2 \left(\left| \sum_{n=0}^{s+I} a_n R^n \right| + p3 \right)$$

The second inequality ensures that destructive cancellation does not render the sum useless. If k = o the first inequality expresses the fact that the last I terms of the series do not significantly affect the sum of the series.

These two conditions are normally sufficient. However it is possible for a series to contain blocks of zero terms.

e.g. $y'' = x^3 y$

has a series expanded about x = o of the form

$$a_0 + a_1 x + 0.\, x^2 + 0.\, x^3 + 0.\, x^4 + a_5 x^5 + a_6 x^6 + 0.\, x^7 + \ldots$$

Although it is sufficient to examine the contribution to the sum from I terms, they are not necessarily the last I terms. **If k is** the number of zero terms in each block, i.e. 3 in this case, and the sum of the last k + I terms is calculated, the non-zero terms are sure to be included.

This "gap" may not remain constant along the series, but it does not increase. Consequently as k is determined at the beginning of the series the inequality will certainly be strong enough.

The constant p3 serves to make the conditions reasonable when the function passes through a zero.

Normally the constants p1, p2, and p3 are automatically set to 10^{-7}, 10 and 1 respectively. They may however be treated as initial conditions and reset to different values

e.g.
range
p1 = 1, - 5
p2 = 1000
x = 0 (1) 10 etc.

This would make the integration process less accurate but would produce results in less time.

When the practical radius of convergence has been found the series are analytically continued and expanded about the new base. In this manner each tabulation point is reached.

OPTIMISATION OF NUMBER OF TERMS

At each analytic continuation the quantity

$$T = \frac{n^2 + p4\ n + p5}{r(n)}$$

is calculated for n = s and n = s - 1, where s is the number of terms in the series and r is the radius of convergence. The expression

$$n^2 + p_4\ n + p_5$$

is proportional to the time taken to calculate a series of n terms. Consequently, if a range R has to be traversed, the time taken will be proportional to T.

If it is found that $T(s - 1) < T(s)$ then s is replaced by s - 1. Otherwise s is replaced by s + 1, subject to an upper limit.

By adjusting the number of terms in this way the integration time is kept to a minimum.

The constants p_4 and p_5 are normally set to 4 and 50 respectively but may be reset.

SECTOR 2

As R4440.4 is read into the machine the contents of S2 are transferred to S64 and may be replaced later by setting key 4 and pressing I.T.B. Thus, although S2 is used, neither Pig 2 nor Mercury Autocode need be overwritten.

FAULT NUMBERS AND USE OF HANDSWITCHES

A user familiar with conventional coding will notice the similarity between fault numbers and the fault/error numbers produced by Pig 2, and also in the use of the I.T.B. and the handswitches whose functions are summarised below.

H.S. = o	read a new program (start clear)
Key 0 up	read new initial conditions. If key 0 is already up when the end of a calculation is reached a new set of initial conditions will be read automatically (start no clear)
Key 2 "	read R4440.4 and binary restart tape (teleinput)
Key 3 "	punch binary restart tape (teleoutput)
Key 4 "	restore original S2 (rescue)
Key 7 "	causes the print instructions to be obeyed once. If key 7 is up during the calculation the print instructions are obeyed each time the series ~~is~~ are calculated. (post-mortem long numbers)
Key 8 "	re-integrates the equation (restart)

FAULT NUMBER	NATURE OF FAULT
* 1	Division by zero.
2	Too many instructions (i.e. the equations are too long).
3	Spurious character, not in equations.
* 4	Exponential of too large a number.
5	Directives yn → n not in order.
6	Too many variables.
7	s directive written after y directive.
* 8	Radius of convergence → o.
* 9	Coefficient in series too large.
10	Initial condition omitted.
* 12	Square root of negative number.
* 14	Logarithm of negative number.
15	Directive not set, or incorrectly set.
32	Spurious character or incorrect form of equation.

* encountered during the calculation.

A. Gibbons, Manchester University,

February 1961.