DA-1

PROGRAMMING MANUAL

Accessory DA-1 permits the Bendix G-15 Computer to operate directly as a Digital Differential Analyzer

QUICK-REFERENCE DATA-FINDER

SUMMARY OF CONTENTS

MINOR REVISION

This edition, APR-03581-3, June 1961, supersedes T5-2, March 1958. The changes are:

Page	
10	Physical specifications, 2nd paragraph
29	Empirical functions
64	Table VII, Deletion of Graph-Follower

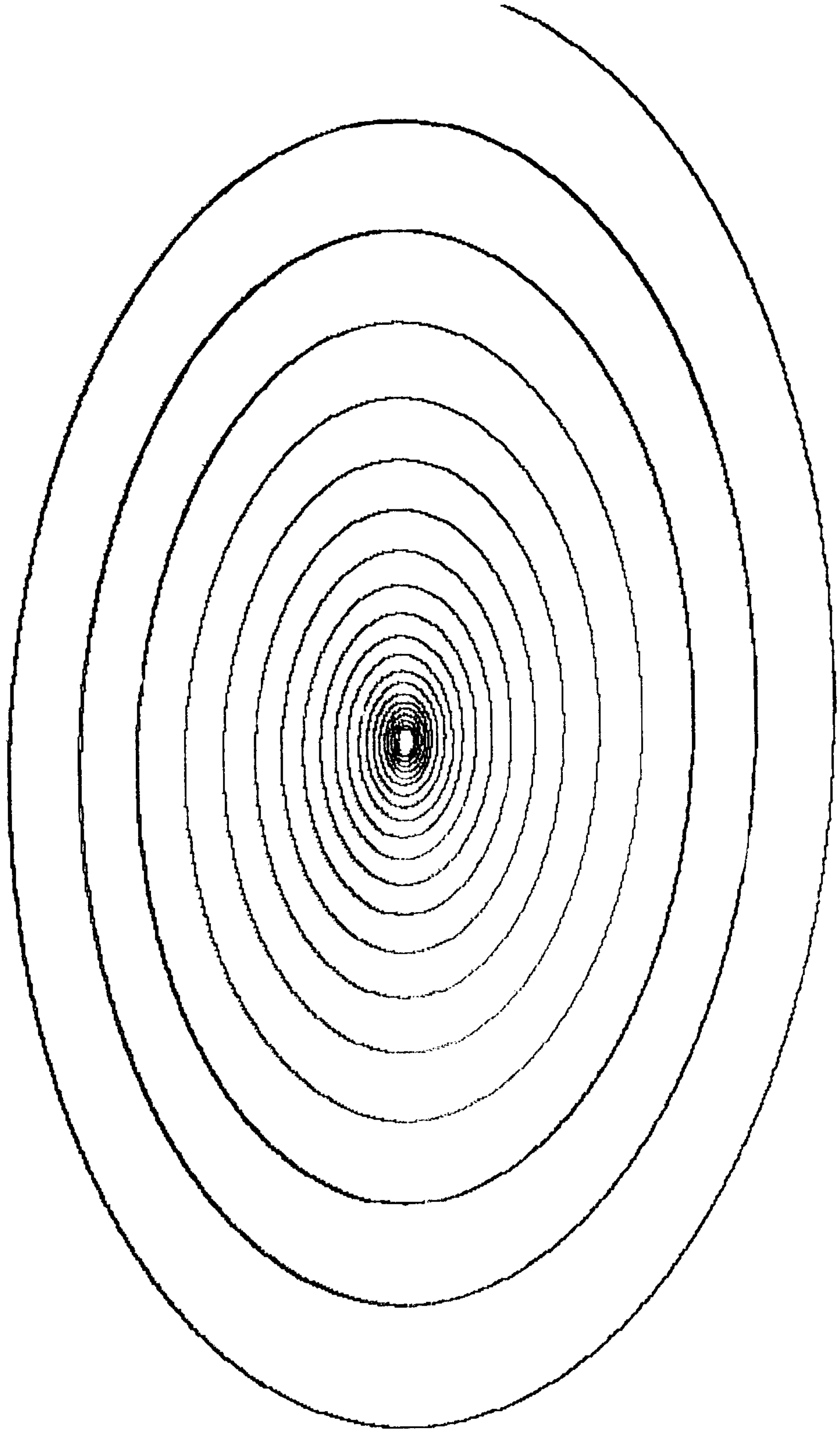


Figure 1

Plot of an elliptical spiral generated with accessory DA-1. Four integrators were required: two to generate the curve, and two to control the graph-plotter. The simple program is described in the text.

PART I - WHAT THE DA-1 IS AND DOES

Purpose of Accessory

The Digital Differential Analyzer Accessory, Model DA-1, permits the simple programming of the numerical solution of problems which can be expressed in the form of ordinary differential equations. These include the solutions of linear and non-linear ordinary differential equations, linear and non-linear simultaneous equations, solutions for roots of transcendental equations, and the simulation of real systems.

In order to solve differential equations on a general purpose computer without a differential analyzer attachment, it is necessary to transform the equation into suitable arithmetic form by use of techniques of numerical analysis and then to formulate coded instructions for the computer. The programming time required is extensive and the programming knowledge required is specialized and often sophisticated. By use of Accessory DA-1, the differential equations, in a slightly altered form, may be placed in the computer and their digital solutions realized. Since the solution is digital, much greater accuracy may be obtained than is possible from earlier-designed differential analyzers which are of an analog nature.

Background

The basic principles of a differential analyzer for the solution of differential equations are not new - they were conceived by Lord Kelvin 100 years ago, described by Sir William Thomson 80 years ago, and put into practical mechanical form by Dr. Vannevar Bush 30 years ago. But the means of execution embodied in the DA-1, the use of digital techniques and the association with a general purpose computer, permit an accuracy, versatility, and ease of use which greatly extend the range of usefulness of this type of machine.

The basic computing unit in the differential analyzer is the "Integrator". An integrator may be thought of as a physical unit (electronic digital registers in the DA-1; mechanical shaft assemblies in the early Bush machine) which has two input lines and one output line. The units may be interconnected so that the output lines of some integrators become the input lines of other integrators. The inputs are differentials of dependent and independent variables arising from the differential equation being solved; in this manual they are called dy and dx, respectively. The output of the integrator is a differential, dz, so that

$$dz = y\,dx$$

The value y, which is the accumulated value of the dy inputs, is held in a register within the integrator. Therefore, if the output line of the integrator is connected to the dependent variable input line of a second integrator, we may obtain in the second integrator

$$z = \int dz = \int y dx$$

Since differential equations usually involve constants, the output of an integrator may be multiplied by any arbitrarily chosen constant value before being used. We may, then, obtain with two integrators

$$dz = K\,ydx \qquad \text{(Equation 1)}$$

$$z = K \int ydx \qquad \text{(Equation 2)}$$

By programming interconnection of integrators of this nature and by selection of initial values for the "y" term in each, any ordinary differential equation may be solved.

Method of Solution

A frequently-used approach is to isolate the highest-order derivative in terms of the other variables. The existence of this derivative in an integrator is assumed and by repeated integration is used to provide all the lower-order derivatives. The lower-order derivatives are then combined according to the conditions of the equation to find the highest-order derivative. The resulting closed loop may then be put in the computer in coded form and solved. An illustration of the technique is given in the example below.

Example

The equation to be solved is:

$$\frac{d^2v}{du^2} + .297\,uv^2\,\frac{dv}{du} - u^2v + \sin u = 0$$

Isolating the highest order derivative:

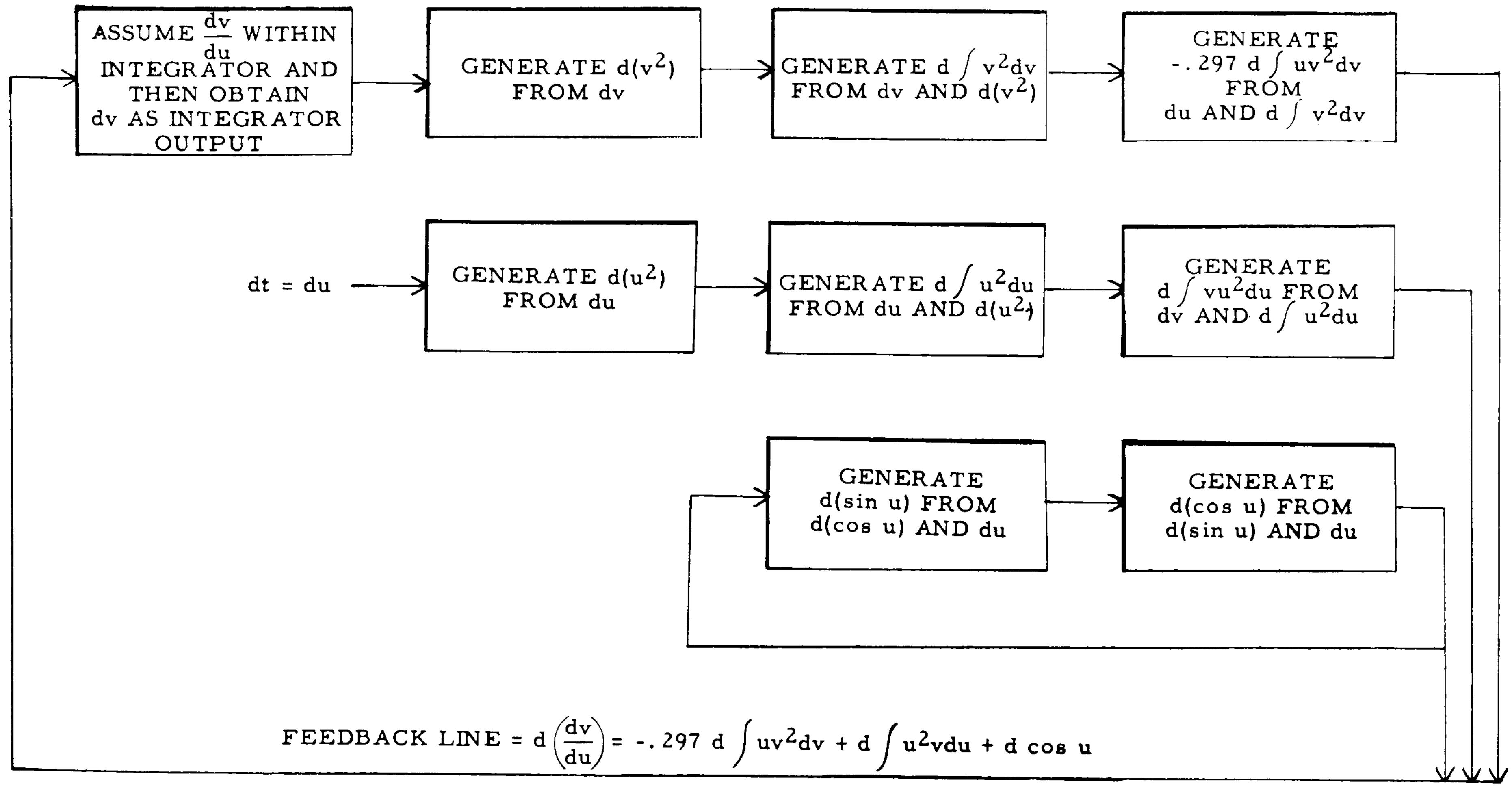
$$\frac{d^2v}{du^2} = -.297\,uv^2\,\frac{dv}{du} + u^2v - \sin u$$

By integrating once with respect to u:

$$\int \frac{d^2v}{du^2}\,du = -.297 \int uv^2\,\frac{dv}{du}\,du + \int u^2v\,du - \int \sin u\,du$$

or

$$\frac{dv}{du} = -\,.297 \int uv^2\,dv + \int u^2v\,du + \cos u$$



METHOD OF SOLUTION FOR EQUATION $\frac{d^2v}{du^2} + .297\ uv^2 \frac{dv}{du} - u^2v + \sin u = 0$

EACH BLOCK REPRESENTS ONE INTEGRATOR

Figure 2

This equation may be put in differential form:

$$d\frac{dv}{du} = -.297\, d\int uv^2\, dv + d\int u^2 v\, du + d \cos u \quad \text{(Eq. 3)}$$

Since the inputs and outputs of integrators are differentials, Equation 3 now describes a relationship between integrators. A machine independent variable called dt is available in the DA-1. This variable provides du. $\frac{dv}{du}$ is assumed to exist in the y register of one integrator. By development from $\frac{dv}{du}$ and du, the right-hand terms in the equation may be obtained as the outputs of integrators. These outputs may then be combined in the manner of Equation 3 to form $d\frac{dv}{du}$. By making $d\frac{dv}{du}$ the dependent differential input into the integrator in which $\frac{dv}{du}$ was assumed to exist, the feed-back loop is closed and the layout of the solution is completed.

The procedure is summarized in block form in Figure 2. Each block represents one integrator and each functions in the manner of Equations 1 and 2. The solution of this problem is described in detail on page 31.

The diagram of the solution is then put in coded form and inserted into the computer. By use of a standard program preparation routine, the initial values of the "y" terms in the integrators and the values of the constant multipliers may be expressed as decimal numbers. The interconnection paths of the integrators are specified numerically in the program code.

Since Accessory DA-1 provides 108 integrators and 108 constant multipliers (a much greater number than heretofore available in digital differential analyzers), it may be used to obtain numerical solutions of complex problems involving high-order differential equations or sets of simultaneous equations.

Characteristics

Distinctive features of the DA-1 are listed below.

108 integrators and 108 constant multipliers are provided.

The primary functions of an integrator are those described in Equations 1 and 2; but, an integrator may also be used as a servo, as a decision element, or for the addition of variables.

The outputs of all integrators are available to be used as inputs by every integrator. Interconnection is by numerical coding and not by patchboard.

During a single step of computation, an integrator receives dy and dx inputs and accumulates both dy and ydx products. Whenever the sum of ydx products exceeds a pre-assigned quantity, the integrator has a dz output. Accuracy and speed of computation has been increased by making the dz output ternary rather than binary: dz may assume a value of + one increment, 0, or -one increment.

An integrator may be coded to receive a dx input from one other integrator; it may be coded to receive dy inputs from any number of other integrators. The value of the dy input during one step of computation must range between -7 and +7 increments.

The set of 108 integrators is processed 34 times per second. The speed of computation is independent of the number of integrators used in the program. The speed of solution of a problem is inversely proportional to the degree of precision required.

The general computing facilities of the G-15 may be used in conjunction with the DA-1 for the programming of highly involved problems.

Empirical functions may be used.

A standard program preparation routine provides the following input-output characteristics:

Initial values of integrands and values of constant multipliers are expressed in decimal form as positive or negative floating decimal point numbers. Precision of computation remains optional. Up to 7 decimal digits of precision is available. Exponent range is from 10^{-4} to 10^{+9}. Computed results are typed out as floating point decimal numbers.

Physical and Electrical Specifications

The DA-1 is controlled from the G-15 General Purpose Computer. The DA-1 has no controls of its own and may not be used independently.

Input can be programmed from normal G-15 equipment. This includes typewriter, punched paper tape, punched cards and magnetic tape. Empirical function input may be from any G-15 input device. Output may be tabulated by normal G-15 equipment or plotted graphically.

The graph-plotter plots the relationships between any two variables generated by the computer in 0.01 inch increments on a standard, sprocketed roll of paper, 1 foot by 100 feet. It is an optional accessory to the DA-1.

The DA-1 is 22 inches deep by 24 inches wide and is 60 inches high. It weighs 300 lbs. Power input is 1 kva, 110-120 volts, 60 cycles, single phase. Cooling is by internal forced air.

PART II - HOW THE DA-1 WORKS

It has been stated that numerical integration can be performed with two integrators. An incremental output, dz, (which is equal to either +1, -1 or zero) is obtained from one integrator so that

$$dz = ydx$$

and the incremental outputs are accumulated in a register in a second integrator in order to obtain

$$z = \int dz = \int ydx.$$

The procedure used may be explained graphically:

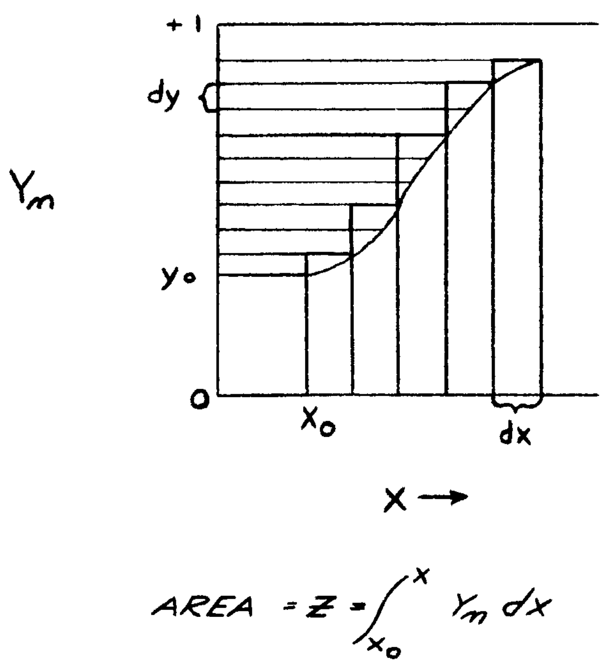


Figure 3

Referring to Figure 3, consider that we are evaluating the definite integral of Y_m with respect to x, by continuously accumulating the area under the curve $Y_m = f(x)$ as x increases.

Initially, the curve is at point y_o, x_o. The curve is approximated by incremental changes in y and x. The incremental changes are called dy and dx on the diagram. If Y_m is the instantaneous value of y, each time x increases by one dx increment, a rectangle of width dx and of height Y_m will be added to the accumulated area. The incremental changes in Y_m, which may consist of from one to several increments at a time, will be rounded off in a manner which reduces the triangular error made at the tops of the rectangles. The method of this round off will be discussed later. Although we must keep track of the value of Y_m in order to know how high a rectangle to add with each dx change, we need not keep track of the entire accumulated area. We may simply send out notification each time an increment of area has been covered and allow the total area to accumulate elsewhere. It will then be necessary to keep track of only Y_m and of any fractional part of an increment of area. In Figure 3 we will consider that a rectangle of width dx and of height +1 (the full height of the diagram) will be our increment of area represented by a dz pulse.

It is in this manner that the DA-1 integrator operates. Each integrator, which is represented by a block as shown in Figure 4, receives two incoming lines of information. One of these, the dx line, notifies the integrator each time that x has increased or decreased one increment. The dy line notifies the integrator of

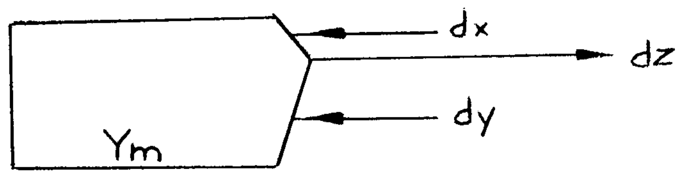


Figure 4

the corresponding changes in Y_m. The output of the integrator is the dz line through which the integrator signals each time an increment of area has been traversed. If the dz line is fed into another integrator as its dy input, this second integrator will then contain as its y value the total instantaneous value of the area Z.

It is necessary for the integrator to contain two registers (see Figure 5). One, called the Y register, algebraically accumulates the value of Y_m. The other, called the R register, contains the fractional part of

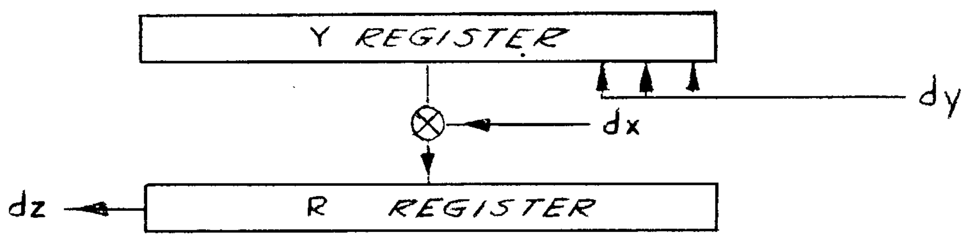


Figure 5

increments of area; the value in the R register will range between 0 and one increment.

The actual digital operation is accomplished in a straight-forward manner. dy increments are added to Y_m algebraically. Each time a dx pulse is received, Y_m is added into the number which exists in the R register. Since neither register is cleared in the process, the R register will periodically overflow. It is the overflow of the R register which causes the dz output signals. Each dz pulse corresponds to the increment of area used in the graphical illustration.

The multiplication of an integrator output by a constant, k, is similarly done by two registers. One, the K register, has no modifying inputs. The other, the K_r register, has the value in K added to it whenever a dz output occurs. The overflow of the K_r register provides Kdz output signals.

Electronically the Y, R, K and K_r registers are parallel memory channels on the magnetic drum of the G-15 computer. Each word-time in the channel(s) represents one integrator. Since there are 108 word-times in a drum channel, we may store information for 108 integrators on the drum.

Consequently, integrators are processed serially, beginning with integrator 00. A single processing of all 108 integrators, in which dz outputs are transferred between integrators, is called an iteration.

As a second consequence, the time required to compute a solution is not a function of the number of integrators used in the program, but is a function of how soon incremental inputs to integrators cause incremental outputs. The amount of variation in the y value, or integrand, to be represented by a single incremental input is optional. Naturally, the smaller this incremented variation in the integrand is made, the greater the precision of the solution. Each additional binary digit of precision doubles the time required to compute a solution.

Round-off Correction

As has been stated, a round-off correction is made in each integrator. The correction consists of starting with one-half full value in each R register. Now a dz overflow will occur from the addition or subtraction of more than one-half full value into the R register.

Therefore, if dz is used as the dy input for another integrator, the resulting digital curve will tend to straddle the average continuous value of z. A tendency to accumulate round-off error is reduced.

Problems solvable with the **DA-1** will take the form of equations. The equations may be algebraic, differential or integral in form, or may occur as simultaneous sets of such equations. The first step in obtaining the solution is to arrange the integrators to generate all of the terms appearing in the equations. The generation of terms by integrators will be considered in this section.

The basic integrator equations (Equations 1 and 2) can be represented schematically. Each pointed block represents one integrator. Each rectangular block represents the constant multiplier associated with the integrator.

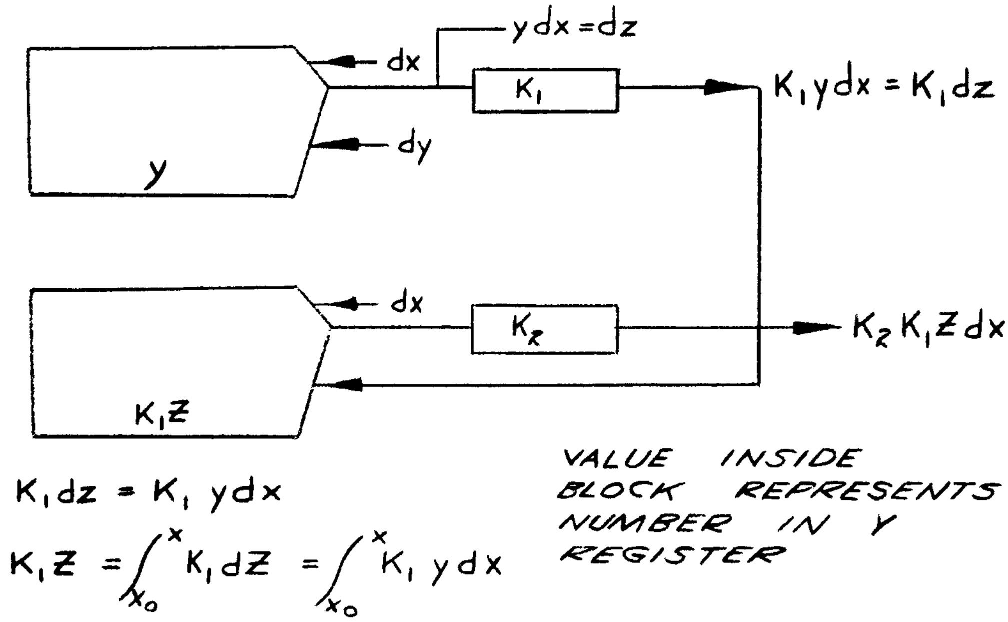


Figure 6

Assume that prior to the receipt of any dy or dx inputs, the Y registers of both integrators were empty. In the first integrator, dy inputs accumulate to form y. A dz output is generated equal to ydx. It is modified by the constant multiplier associated with the integrator to form K_1 dz. The K_1 dz output is sent as the secondary input to another integrator where it is accumulated in the Y register of the second integrator.

Now if at the start of computation the Y register of the first integrator had held the value C_1, and if the Y register of the second integrator had held a value C_2, the output from the first integrator would be

$$Kdz = K_1(y+C_1)\, dx$$

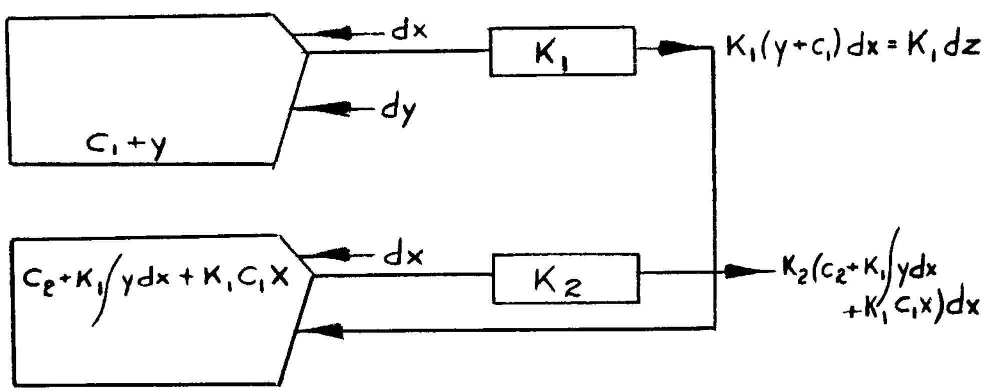


Figure 7

and there would be accumulated in the second integrator:

$$C_2 + K_1z = K_1\int ydx + K_1C_1x + C_2.$$

If the two inputs to an integrator are tied together as in Figure 8:

$$K dz = K(x+C_1)\, dx$$

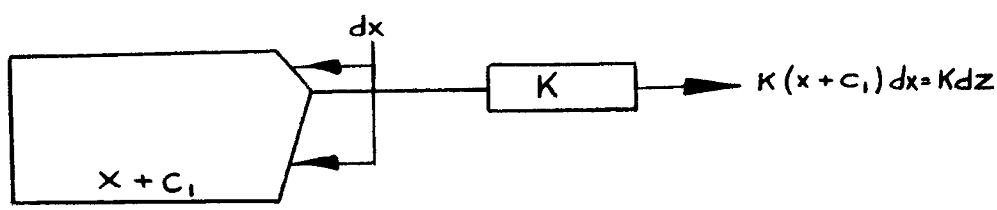


Figure 8

which when accumulated will produce the quadratic function

$$C_2 + Kz = K(x^2/2+C_1x) + C_2$$

In the equation and diagrams in the remainder of this section, the constants will be omitted. These constants, constants of integration, are determined by the initial settings of the integrators.

If the output of an integrator is fed back as its own dy input, as in Figure 9, an exponential function is generated:

$$dz = zdx$$

$$z = e^x$$

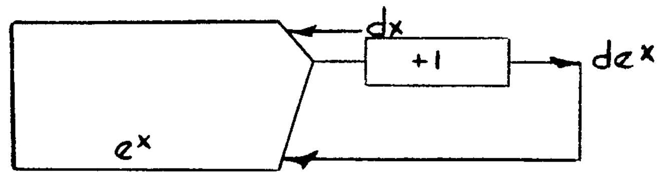


Figure 9

More complicated functions can be formed by causing the Y register of an integrator to accept more than one input. An example is shown in Figure 10.

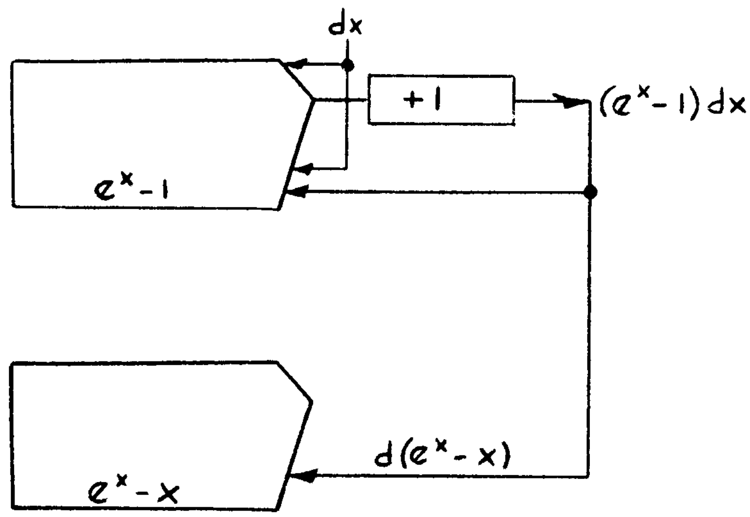


Figure 10

Mapping

The diagrams in both Figures 6 and 7 show pairs of integrators with connections representing information flow drawn between them. The preparation of such a flow diagram to represent the generation of a function or the solution of a problem is called "Mapping". Some examples of maps of functions follow.

The map of a complex problem may require a great many integrators. It is in mapping that the creative aspects of programming for the DA-1 lie. The solution of a problem may usually be represented by more than one map, each map representing an alternate approach. The procedure for mapping a problem is described in detail in a subsequent section.

Generation of Functions

The functions mapped on the following pages are common ones useful in the solution of problems. The types which may be generated internally with Accessory DA-1 include algebraic, trigonometric, exponential, logarithmic, hyperbolic, Bessel and probability functions.

Maps for function generation can be devised by use of a standard table of integrals. For example, the flow diagram in Figure 11 is based on the relationships:

$$\ln v = \int \frac{1}{v}\, dv$$

$$\frac{1}{v} = -\int \frac{1}{v}\, d \ln v$$

The first equation is represented by the first integrator in the map:

$$dy = d(\tfrac{1}{v})$$

$$dx = dv$$

$$Kdz = d \ln v$$

$$K = +1$$

The second equation is represented by the second integrator:

$$dy = d(\tfrac{1}{v})$$

$$dx = d \ln v$$

$$Kdz = d(\tfrac{1}{v})$$

$$K = -1$$

In Figure 12, the term $\frac{1}{v}$ is generated by the same means as in Figure 11. In the third integrator of the example, the relationship is used:

$$dv^n = nv^n \, d \ln v \qquad \begin{aligned} dz &= dv^n \\ dy &= d\,v^n \\ dx &= nd \ln v \end{aligned}$$

We can show this equation to be a form of $de^u = e^u du$:

$$v^n = e^{n \ln v}$$ By definition of a natural logarithm

let $n \ln v = u$
then $v^n = e^u$
and $\ln v = \frac{u}{n}$

$$d \ln v = d\,\frac{u}{n}$$

And by substitution in the equation for dv^n:

$$d(e^u) = \frac{ne^u du}{n} = e^u \, du$$

An interesting exercise for the reader would be to demonstrate the validity of the flow diagram in Figure 13 by similar means.

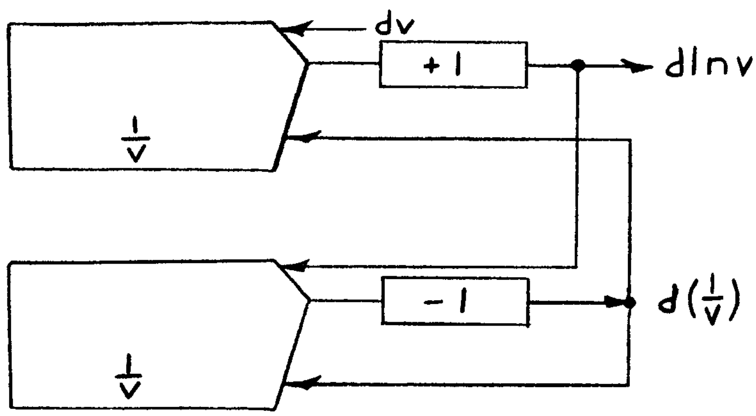


Figure 11

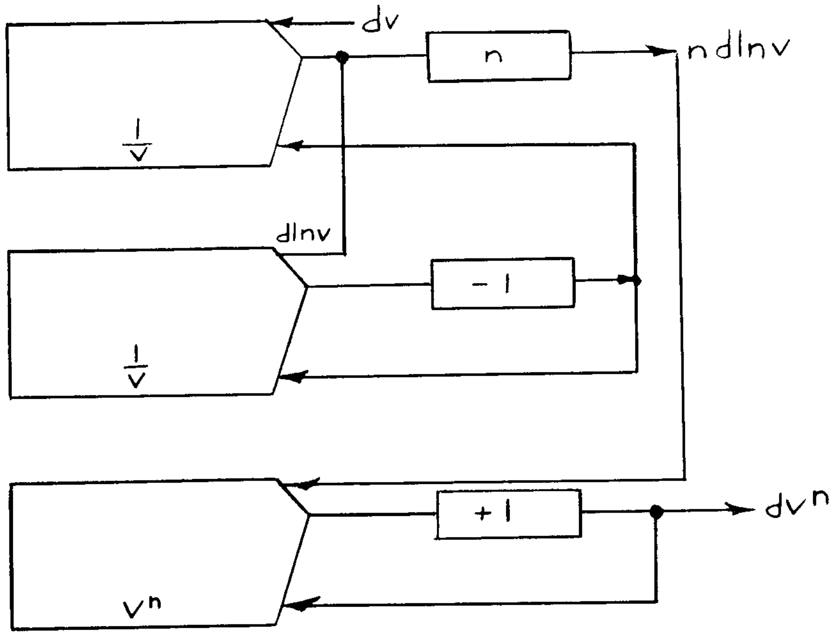


Figure 12

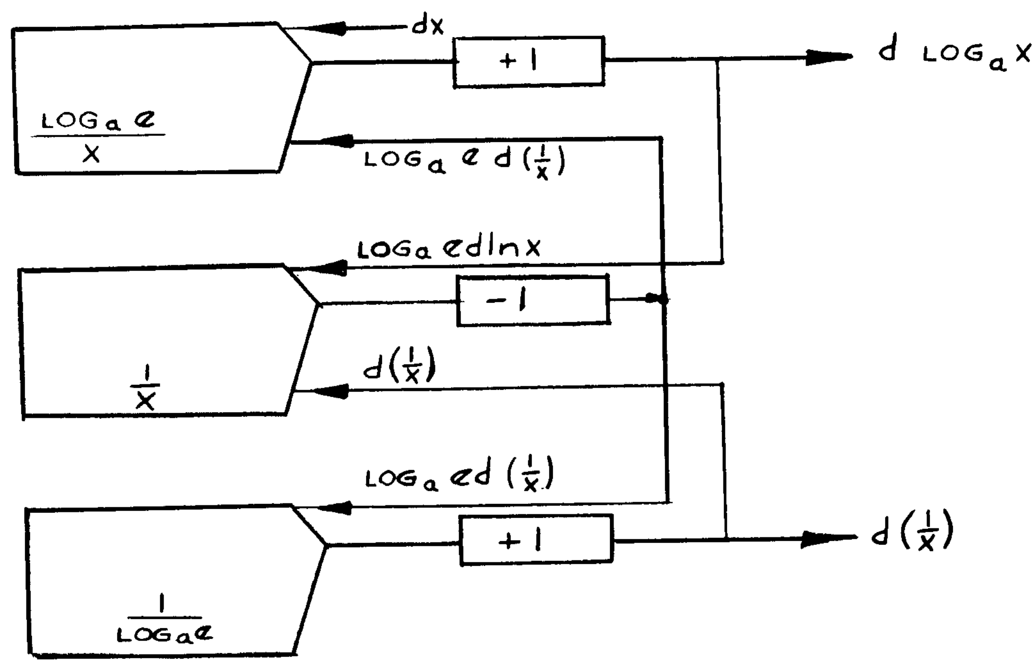


Figure 13

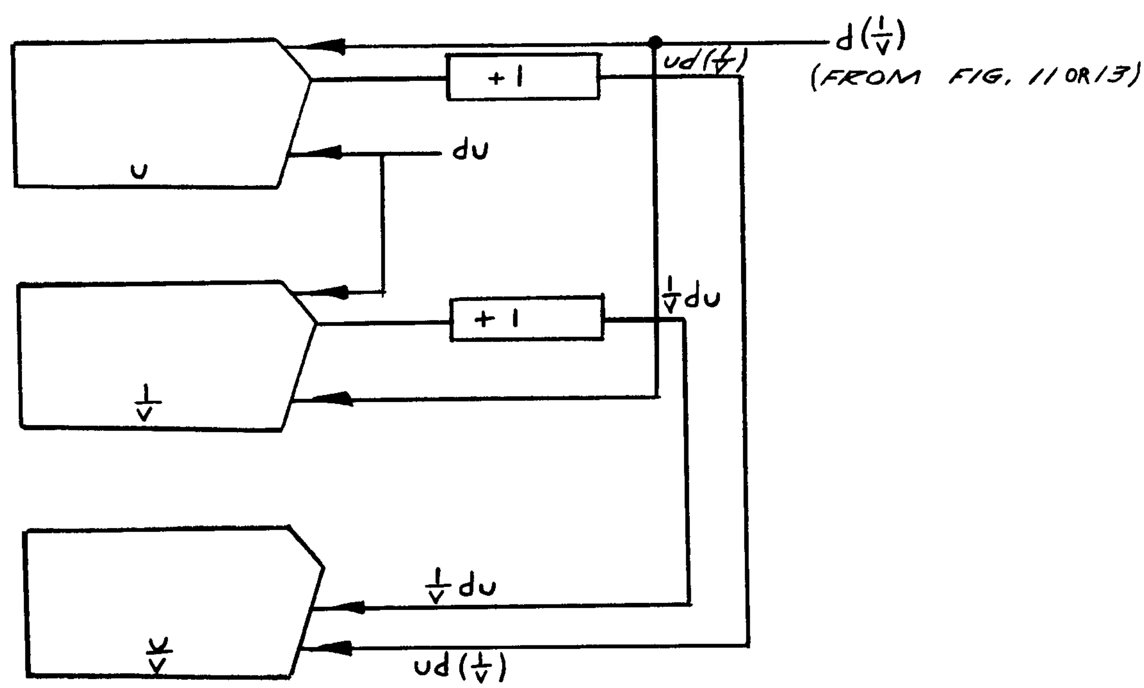


Figure 14

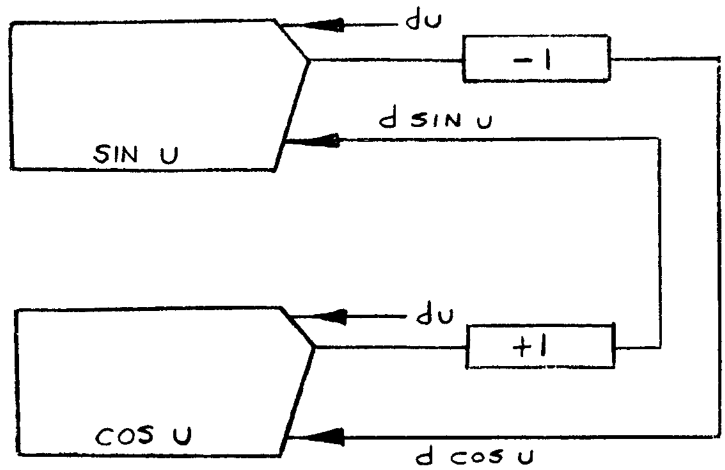


Figure 15

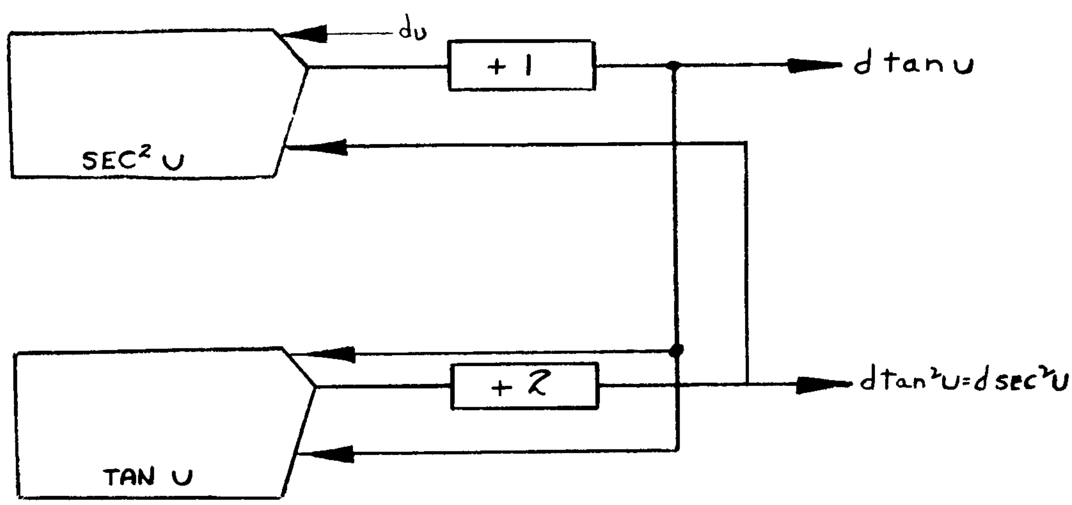


Figure 16

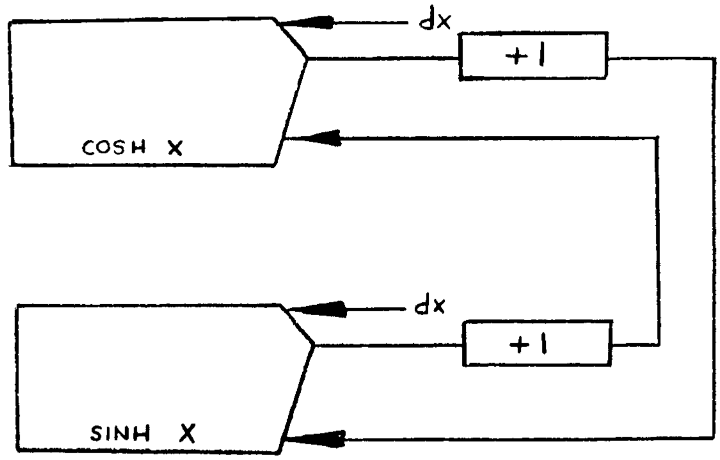


Figure 17

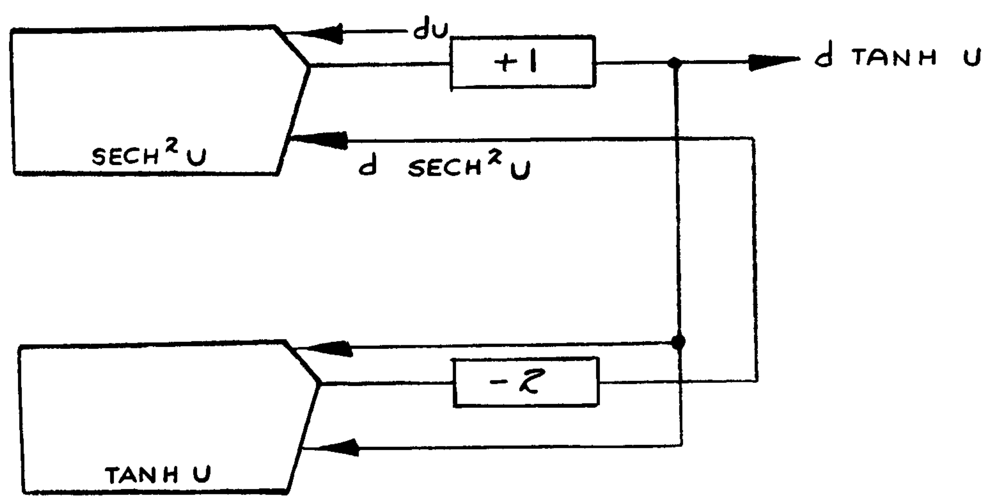


Figure 18

Modes

An integrator may be coded to have either of two modes of operation. The "normal" mode is the one which has already been described: from the dy and dx inputs an output dz is generated so that

$$dz = ydx.$$

dz is equal to either +1 increment, 0, or -1 increment. The output occurs in an iteration in which the sum of ydx products exceeds one increment of scaled output value. Since y may be positive or negative, and dx may be positive or negative, the sum of ydx products may be either positive or negative. The R register holds the sum of the ydx products and the overflow of the R register provides dz.

Decision Operation

In the second mode of operation, the dz output of an integrator is determined differently. The R register is no longer used. Whenever the Y register holds a non-zero value and a dx input occurs, a dz output will occur. The sign of this output will be algebraically determined by the product of the signs of y and dx. Symbolically:

$$\text{If } y \neq 0: \quad dz = (\text{sign of } y)\,(dx)$$

$$\text{If } y = 0: \quad dz = 0.$$

This mode is called "Decision" operation. It is useful in simulating the action of a servo, for the addition of functions, and in handling discontinuities in functions.

Servo Operation

One use of a decision integrator is called "Servo" operation. An integrator coded as a decision element can be made to act as a servo if it is also coded to receive a dx input every iteration. A servo would be used to continually make a correction that tends to reduce an error.

A servo may be used to generate u as a function of v where u is defined implicitly as a function of v by F(u, v) =0. For example in Figure 19, a square root is generated by the determination of u in the relationship

$$v - u^2 = 0.$$

In differential form, this equation becomes

$$dv - 2u\,du = 0.$$

dv and -2u du are made the dy input to the integrator coded as a servo.

When incremental changes in v cause $F(u, v) \neq 0$, the servo generates incremental changes in u until $F(u, v) = 0$.

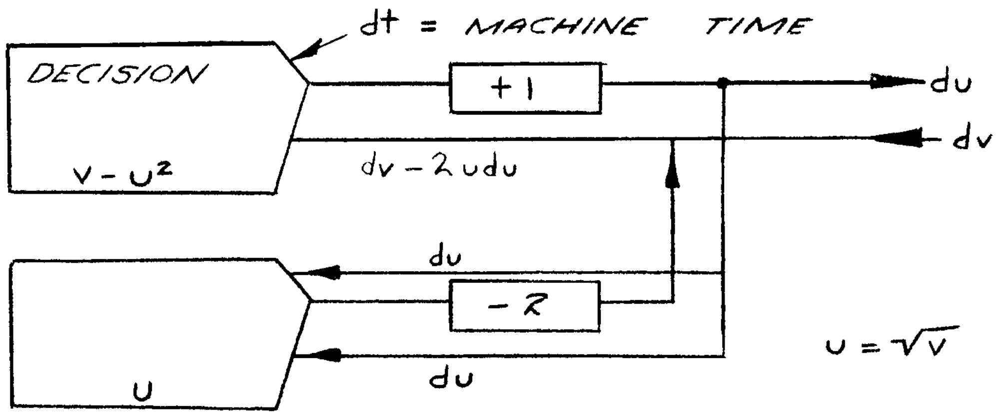


Figure 19

Figure 20 shows a hook-up for the case in which $F(u, v) = e^u - v$.

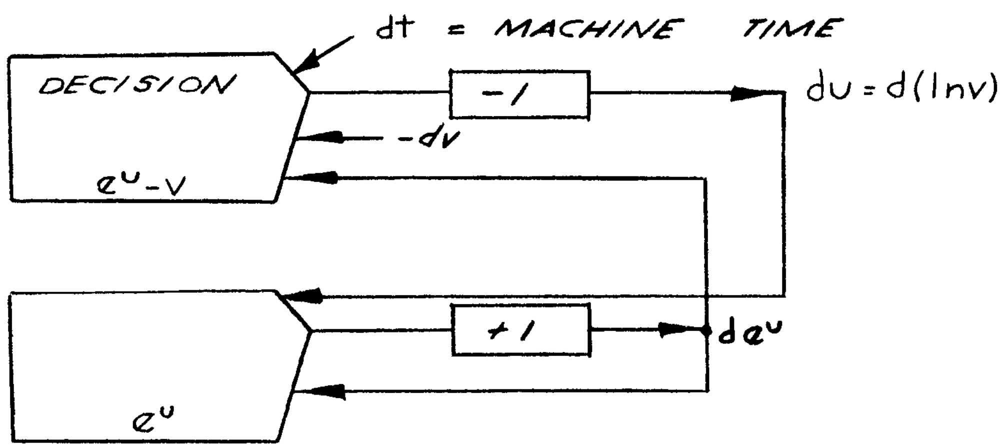


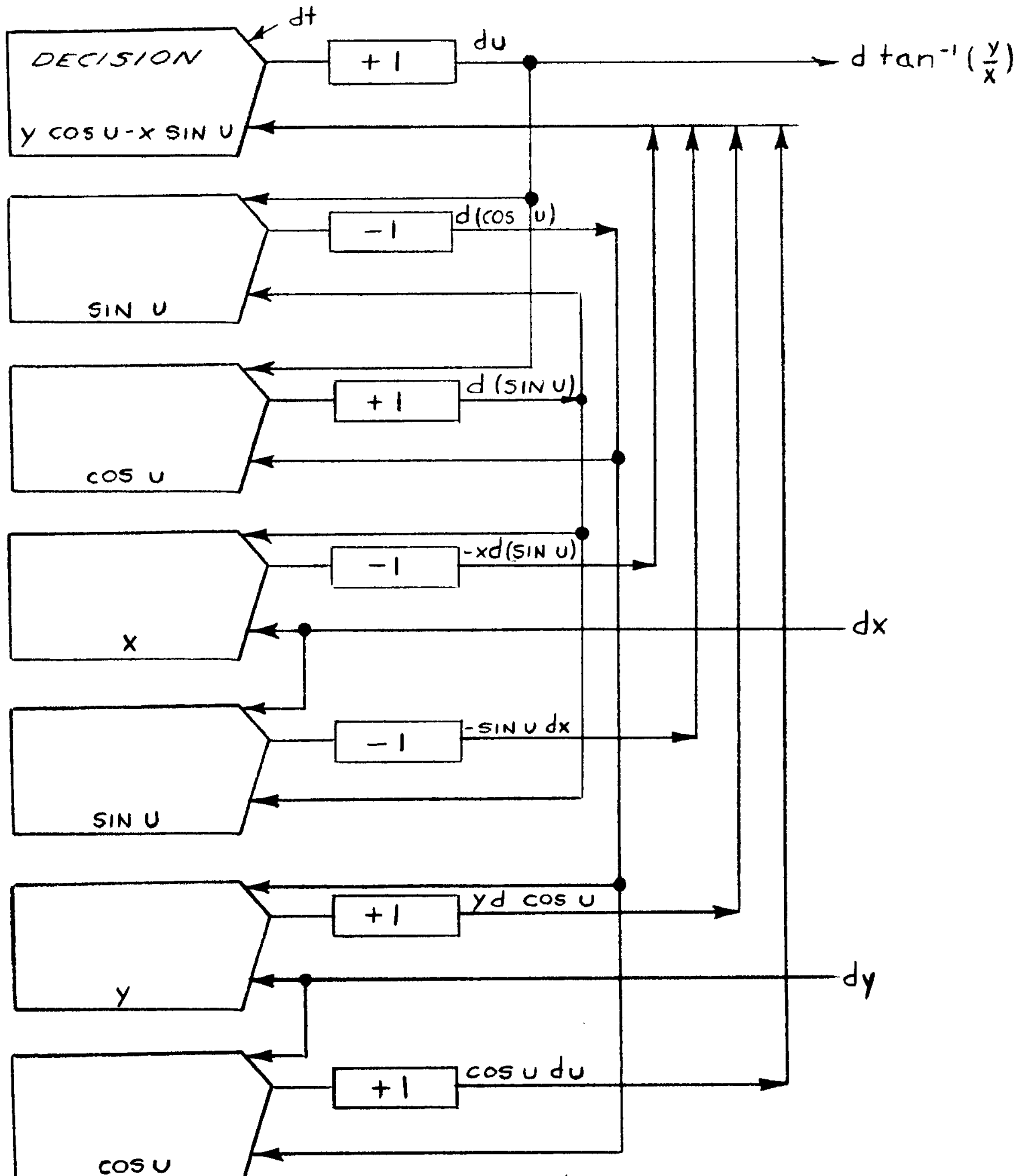
Figure 20

Figure 21 shows the case when

$$F(u, v) = y(v) \cos u - x(v) \sin u.$$

In this latter case $u = \tan^{-1}(\frac{y}{x})$ and

$$\frac{\delta F(u, v)}{\delta u} < 0 \text{ if } |x| + |y| > 0.$$



$U = \tan^{-1} \frac{y}{x}$

Figure 21

Addition of Functions

If dz's from different sources are to be combined as the dy input to a single integrator, the various dz's may be connected directly to the dy input line. The only stipulation is that the total value of the dy input during a single iteration must be between the range of -7 to +7 increments.

If dz's from different sources are to be combined as the dx input to an integrator, they first must be passed through a decision integrator in the manner illustrated in Figure 22. When connected in this manner, the integrator is called an "adder". The Kdz output of the adder is the negative of the sum of the inputs.

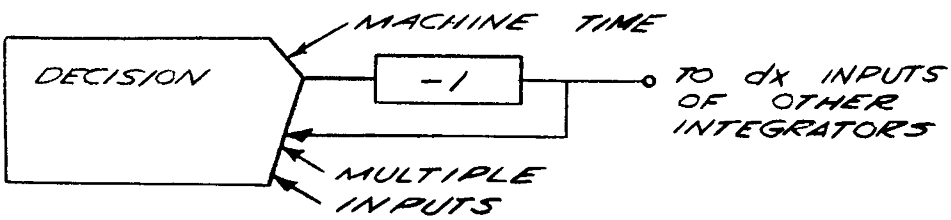


Figure 22
"Adder" Connection for Addition of Functions

Figure 23 shows an adder used to obtain du = +dx +dw. As a result of the servo-like operation of an adder, the function u will be such that x + w - u = 0.

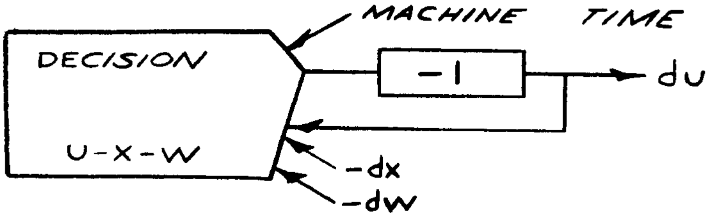


Figure 23

Disconti-
nuities

A decision integrator may also be used to generate a wide variety of discontinuous and non-linear functions, such as square waves, saw-tooth waves, etc.

Figures 24 and 25 show two integrator hook-ups that involve decision integrators. In Figure 24 a decision integrator is being employed to generate the absolute value of a function, u. Since, in any cycle, the same increment is used both as a primary and a secondary input, the decision integrator emits the change in the absolute value of u as its incremental output. In Figure 25 two decision integrators are being used to generate the saw-tooth functions, w and v, by means of same properties.

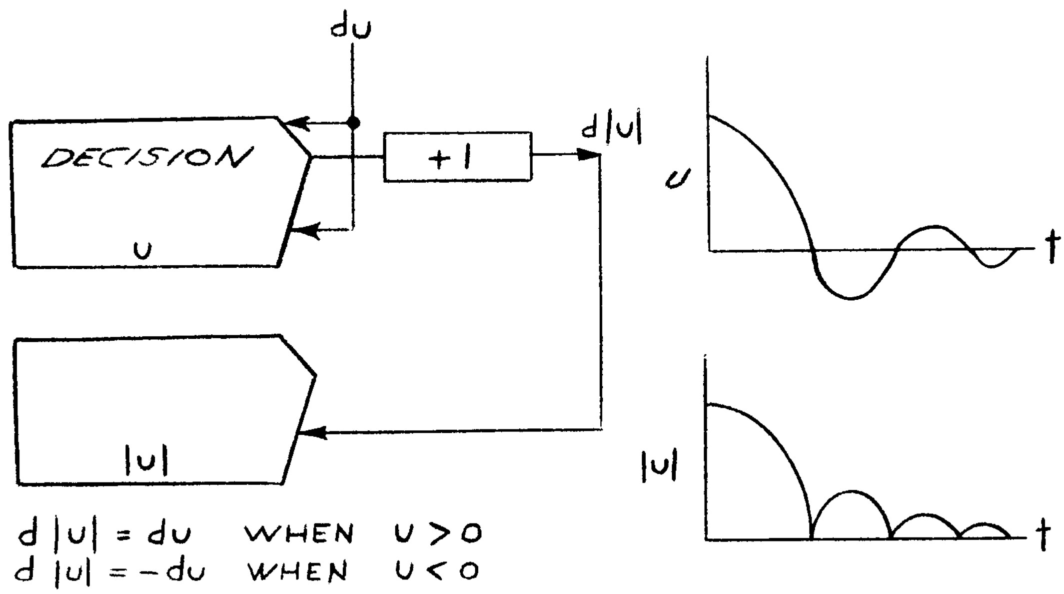


Figure 24

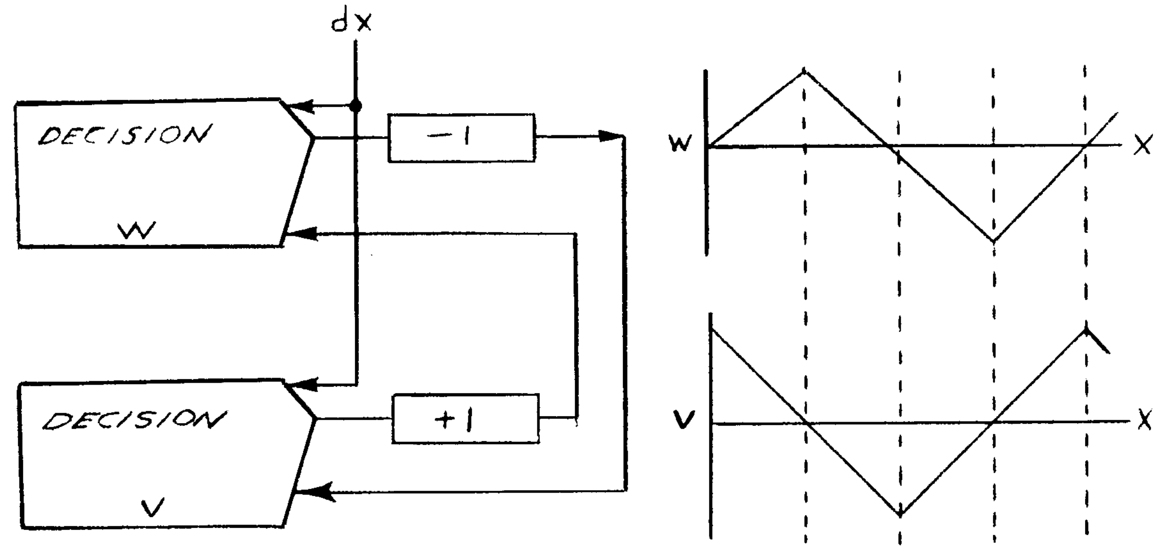


Figure 25

Empirical Functions

Empirical functions may be entered into computations in one of two ways. An empirical curve may sometimes be approximated by the combination of known functions which may be generated on the DA-1. Or, if the empirical data is in numerical form, the integrand values may be sent to the DA-1 via the General Purpose Computer as it is required.

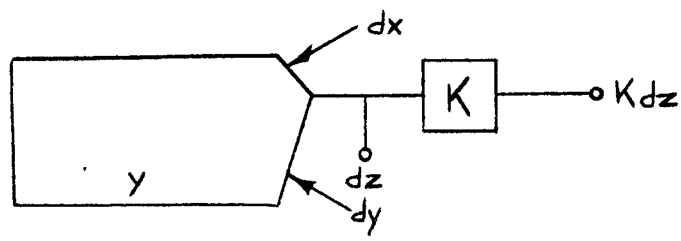
Initial Conditions

A feature of the DA-1 is its ability to remember the initial conditions of the problem being solved. The initial values are determined by the programmer from that particular solution of the equations which corresponds to a desired starting point for computation. (For example, in Figure 3 the point y_0, x_0 specifies the initial conditions.) The initial values, which are stored in the Y registers of the appropriate integrators at the beginning of computation, are destroyed by the computation as the Y register is incremented and decremented. The initial values may be stored in a second set of registers called the Initial Conditions Registers. The contents of these registers do not change during the solution of the problem. At any time in the tabulation of the solution, the computer may be halted and initial conditions reinserted into all integrators. The problem may then be recomputed, with the initial conditions modified, if desired.

PART IV - HOW TO MAP SOLUTIONS

Mapping

There are three steps involved in programming a differential equation for the DA-1. The three steps are mapping, scaling and coding. "Mapping" consists of arranging an integrator information flow diagram, such as those in the preceding section, in a form that corresponds to the problem to be solved.



$$dz = d\left(\int y\,dx\right) = y\,dx$$

$$Kdz = Ky\,dx$$

dz may be sent only to the four consecutive following integrators.

Kdz may be sent to all integrators.

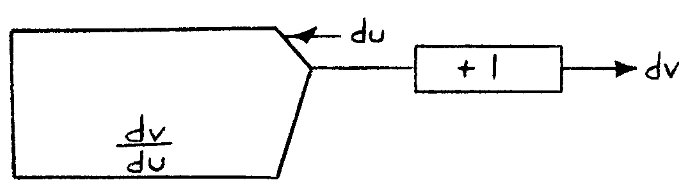
Figure 26

As an illustration of mapping, consider the example discussed in Part I.

After the equation is re-arranged and put in differential form (page 8), we have

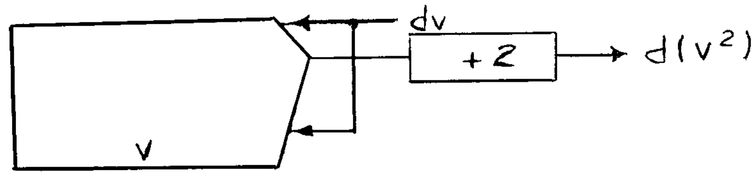
$$d\left(\frac{dv}{du}\right) = -.297\ d\int uv^2\,dv + d\int u^2v\,du + d\cos u \quad \text{(Eq. 3)}$$

If $\frac{dv}{du}$ is assumed to exist within an integrator, the dependent variable dv may be obtained from the output of the integrator:



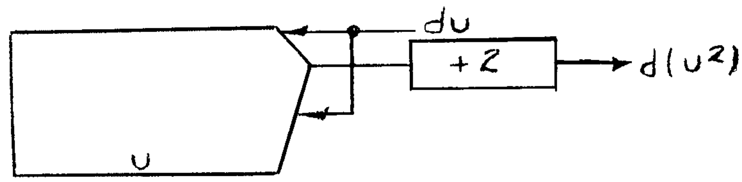
$$dv = \frac{dv}{du}\,du$$

By sending dv to a second integrator as both the dx and dy input, $d(v^2)$ may be generated:



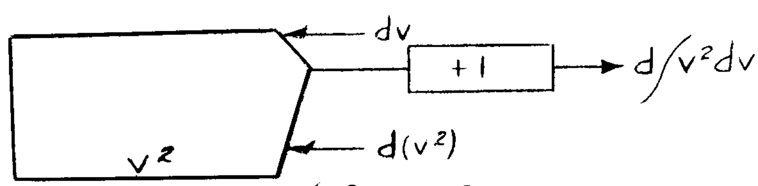
$$d(v^2) = 2v\,dv$$

By similar means, $d(u^2)$ may be obtained from du:

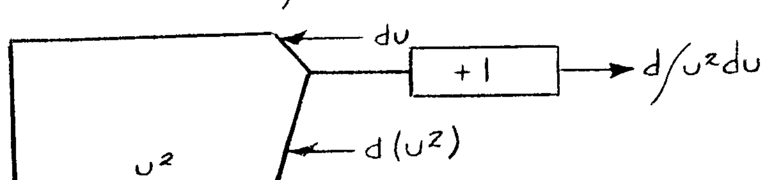


$$d(u^2) = 2u\,du$$

The input terms are now available to generate $d\int v^2 dv$ and $d\int u^2 du$ which in turn are necessary to generate the first two right-hand terms of Equation 3.

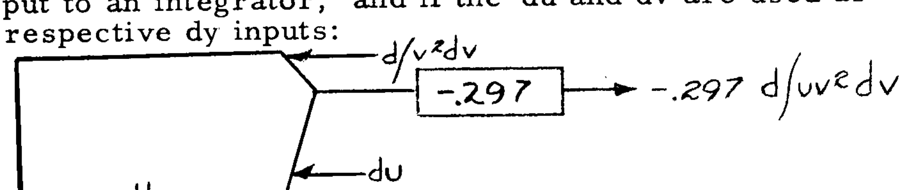


$$d\int v^2 dv = v^2 dv$$

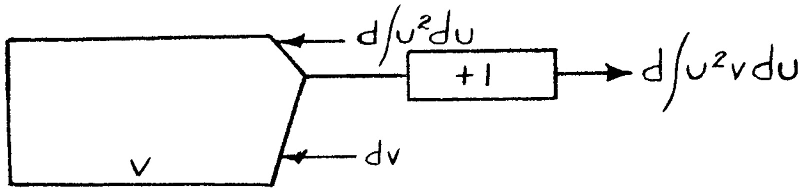


$$d\int u^2 du = u^2 du$$

Now if each of these latter terms is used as the dx input to an integrator, and if the du and dv are used as respective dy inputs:



$$-.297\, d\int uv^2 dv = -.297\, uv^2 du = -.297\, u d\int v^2 dv$$



$$d\int u^2 v\,du = u^2 v\,du = v\,d\int u^2 du$$

The third right-hand term of Equation 3, cos u, is an example of a function which may be generated directly in a differential analyzer. Two integrators are required for the generation by making a closed loop of the relationships.

$$d(\cos u) = -\sin u\,du$$
$$d(\sin u) = \cos u\,du$$

Now $d(\frac{dv}{du})$ may be formed by combination of the other three terms in Equation 3. $d(\frac{dv}{du})$ is made the dependent differential input into the integrator in which $\frac{dv}{du}$ was assumed to exist in order to form $\frac{dv}{du}$.

The completed flow diagram is shown in Figure 27.

As a second example, consider the second order non-linear differential equation:

$$\frac{d^2y}{dx^2} = -\frac{dy}{dx} + y^2 + \sin y + A$$

Three flow diagrams, each representing a different method of solution, are shown for this problem in Figures 28, 29 and 30.

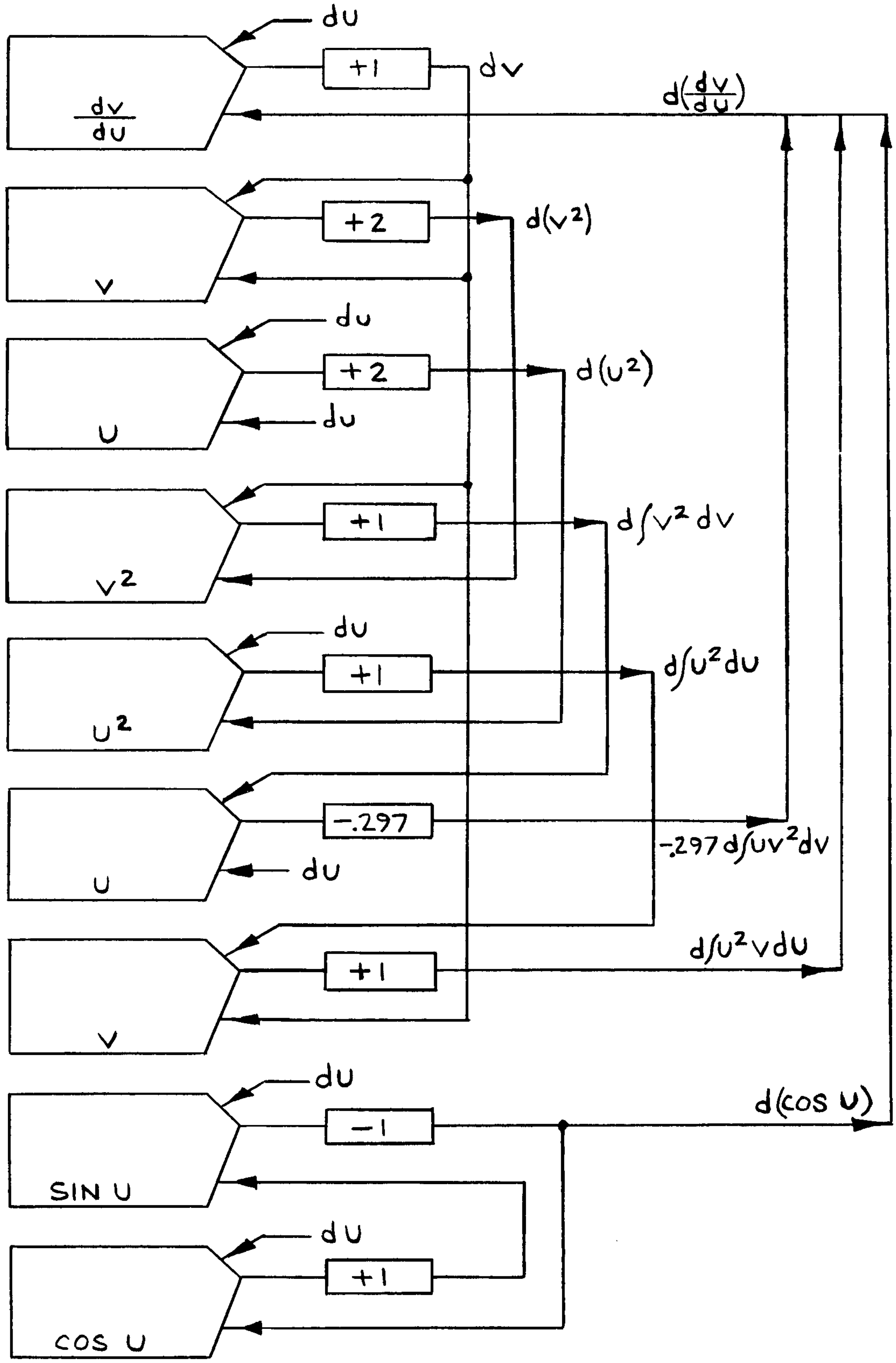

du
+1
dv
d(dv/du)
dv/du
+2
d(v²)
v
du
+2
d(u²)
du
u
+1
d∫v²dv
v²
du
+1
d∫u²du
u²
-.297
-.297 d∫uv²dv
du
u
d∫u²vdu
+1
v
du
d(cos u)
-1
SIN U
du
+1
COS U

Figure 27

METHOD I

Equation $\quad \frac{d^2y}{dx^2} = -\frac{dy}{dx} + y^2 + \sin y + A$

Explanation: The highest derivative has been separated from the rest of the equation. Assume $\frac{d^2y}{dx^2}$ in Integrator 25 in order to form $d(\frac{dy}{dx})$. Accumulate $\frac{dy}{dx}$ in Integrator 28 and generate dy. Accumulate y in Integrator 31 and form $d(y^2)$; Integrators 34 and 35 generate d(sin y). The terms are added together and fed back into Integrator 25 to close the loop.

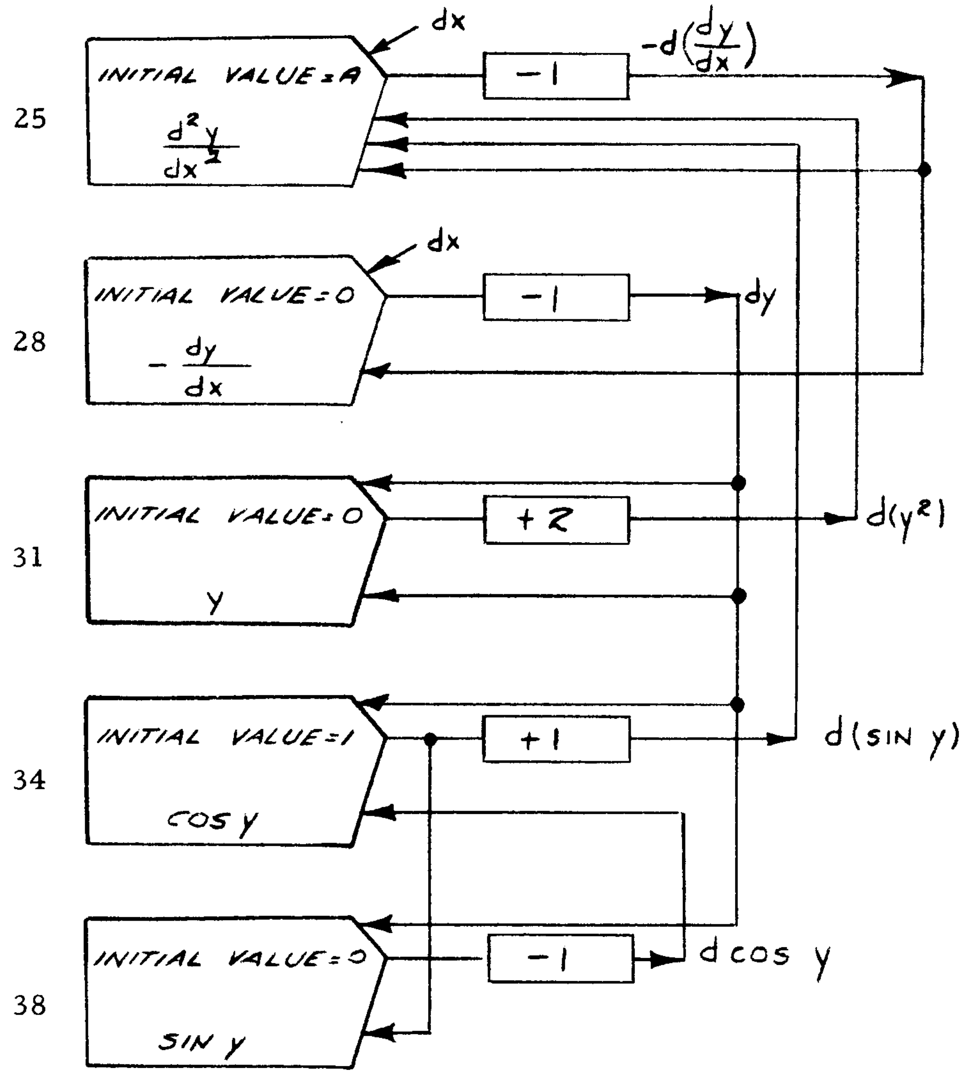


Figure 28

Equation $$\frac{d^2y}{dx^2} + \frac{dy}{dx} - y^2 - \sin y - A = 0$$

Explanation: A decision integrator is used as a servo to control the highest derivative in such a manner as to hold the sum of the terms within one digit of zero. Comparative accuracy is slightly less than that obtained by the first method. This method has the advantage that the equation need not be explicitly solved for the highest derivative.

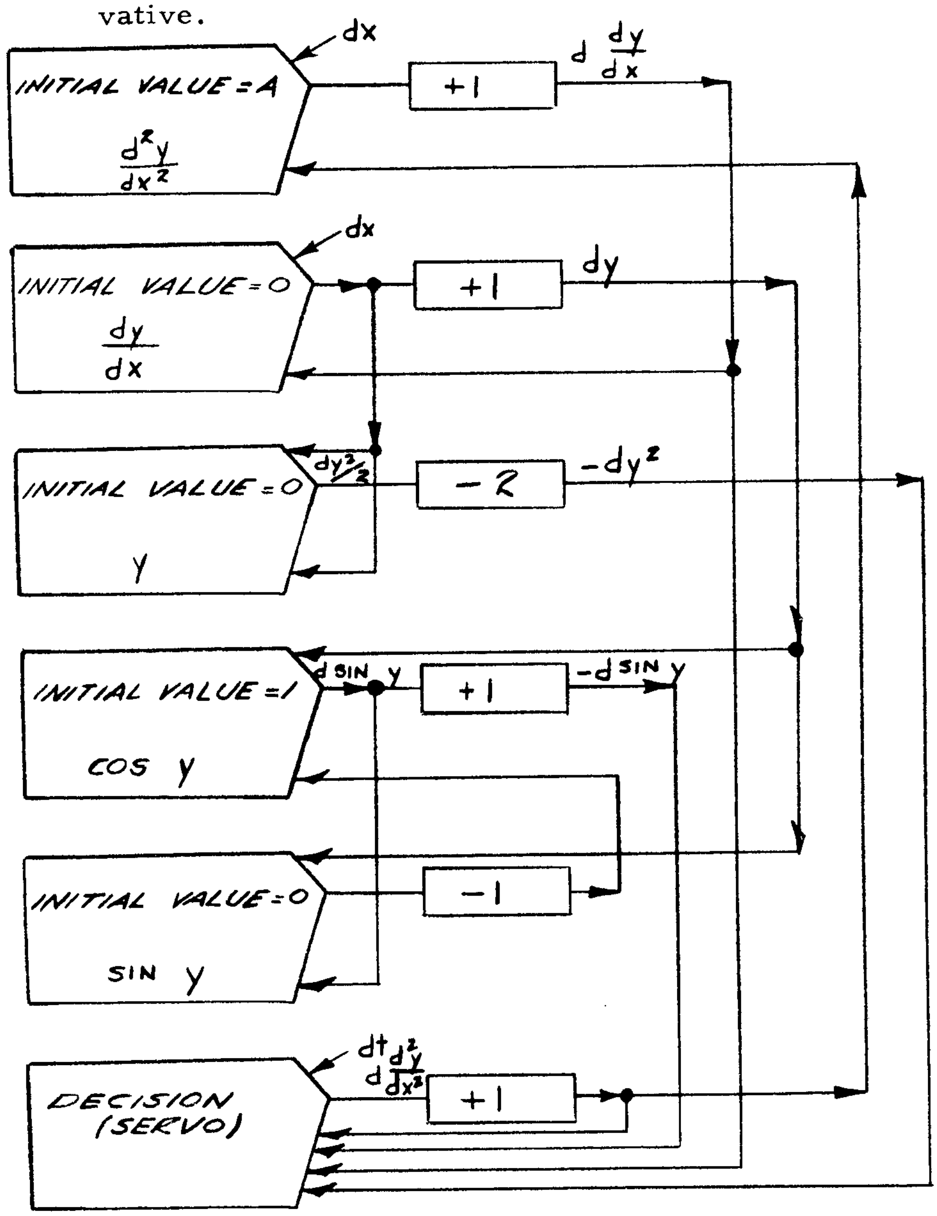


Figure 29

METHOD III

Equation: $$\frac{d^2y}{dx^2} = -\frac{dy}{dx} + y^2 + \sin y + A$$

Explanation: If we integrate with respect to x and then obtain the differential (as in the example in Part I), the terms may all be generated by integrators:

$$d\left(\frac{dy}{dx}\right) = -dy + (y^2 + \sin y + A)\,dx$$

This method has the possible disadvantage that $\frac{d^2y}{dx^2}$ is not explicitly generated in the computation. However, the method layout often conserves integrators.

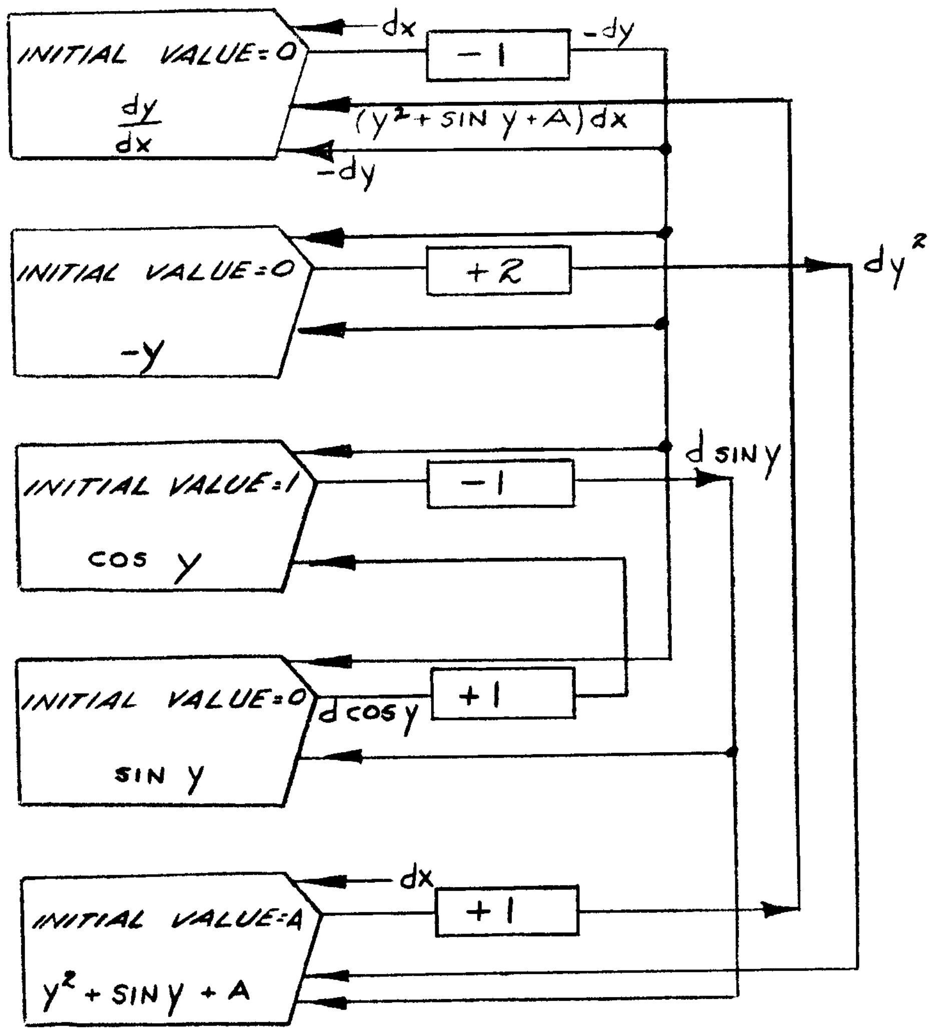
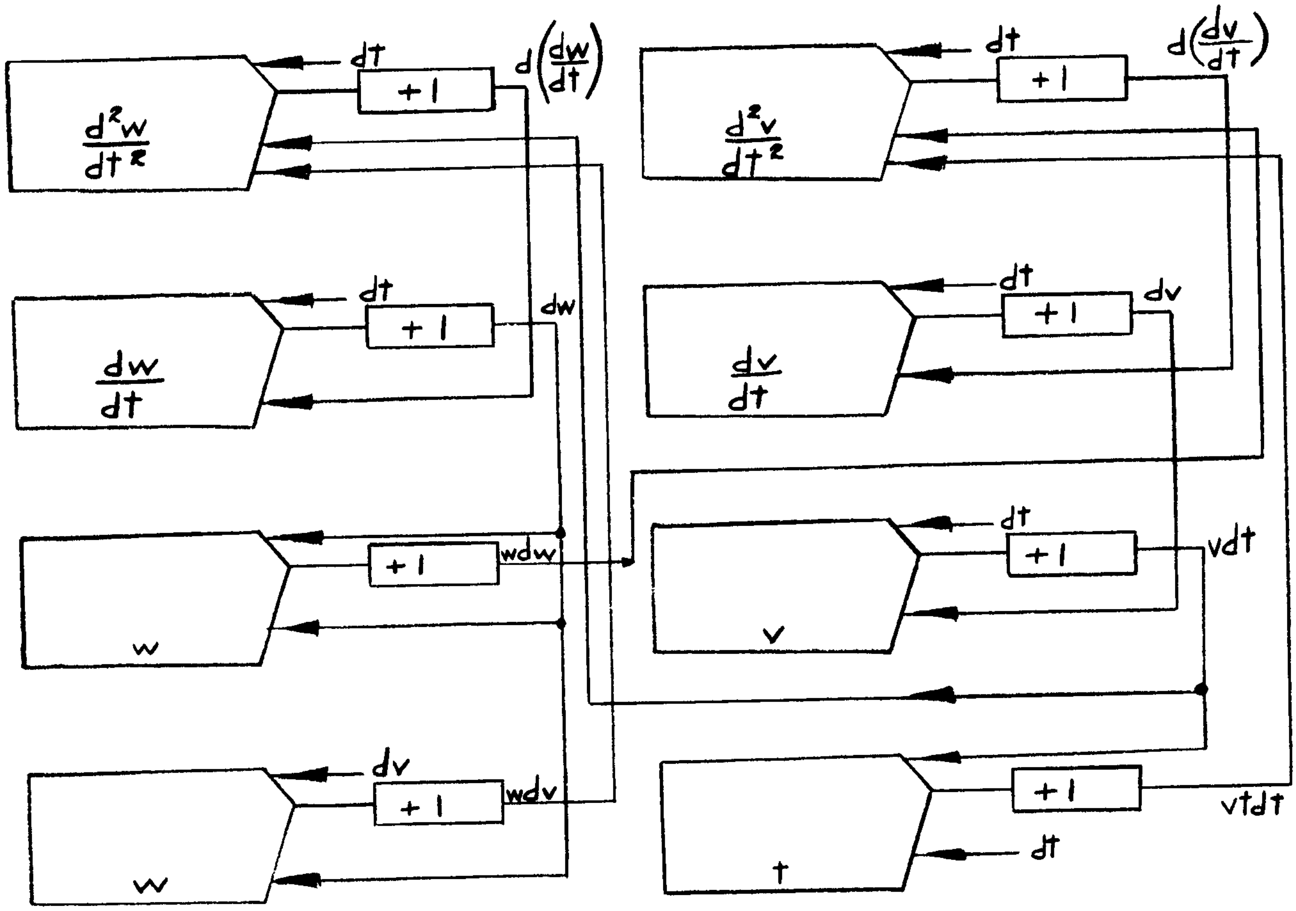


Figure 30

A similar procedure may be used in mapping a problem involving a simultaneous set of differential equations, as shown in Figure 31.

$$\frac{d^3w}{dt^3} = w\frac{dv}{dt} + v$$

$$\frac{d^3v}{dt^3} = w\frac{dw}{dt} + vt$$



$$d\left(\frac{d^2v}{dt^2}\right) = wdw + vtdt$$

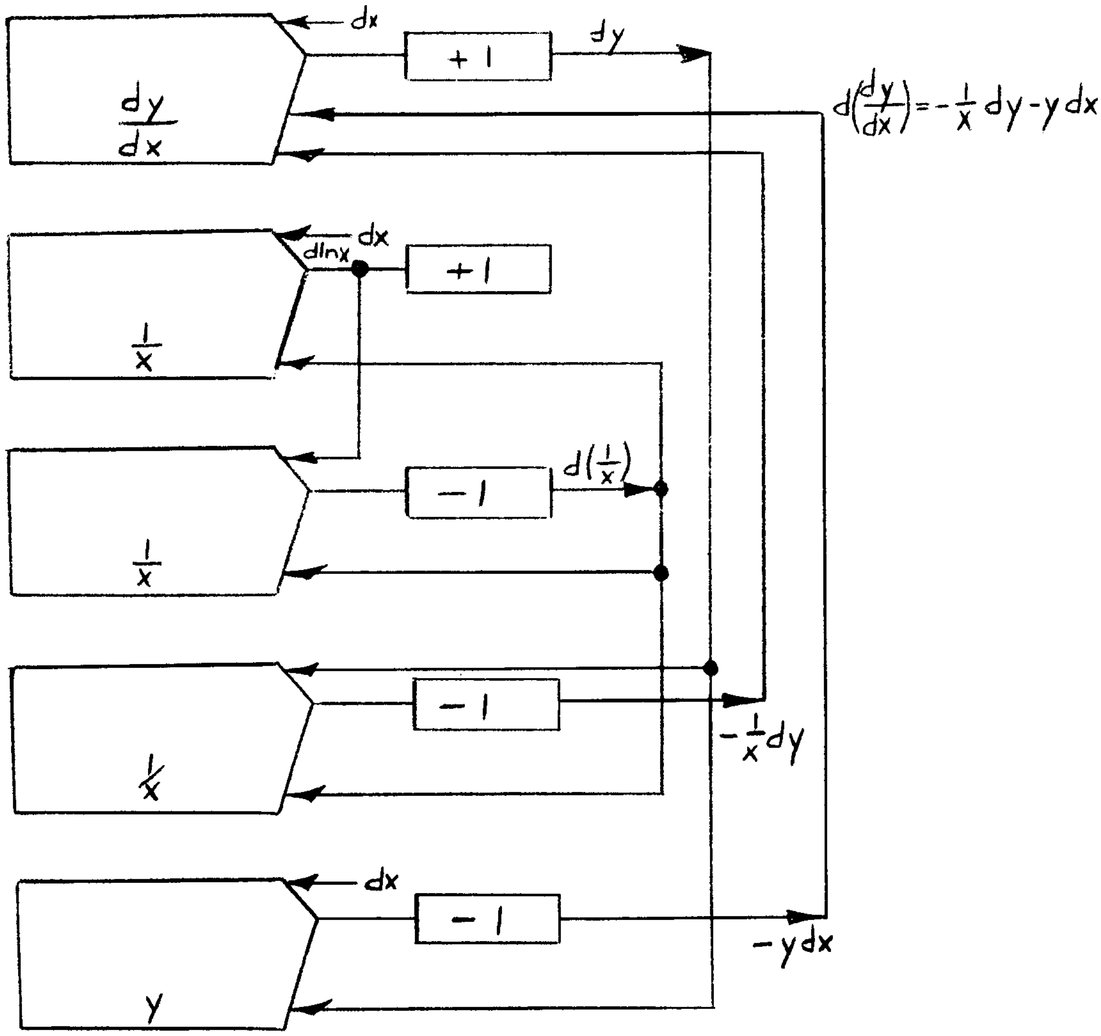
$$d\left(\frac{d^2w}{dt^2}\right) = wdv + vdt$$

Figure 31

It may happen that a particular scheme for mapping will give rise to an integrand which becomes infinite at some point in the problem. For the equation,

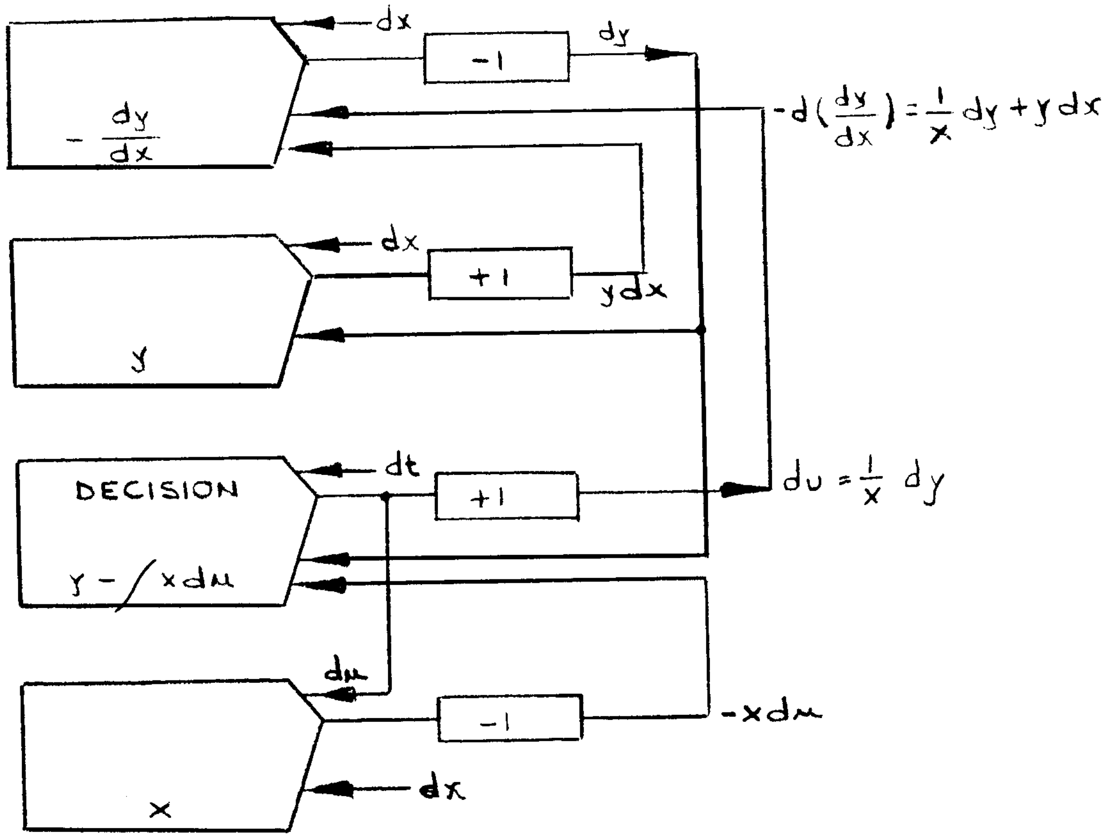
$$\frac{d^2y}{dx^2} = -\frac{1}{x}\frac{dy}{dx} - y$$

the map shown in Figure 32 cannot be used when x is small or zero. Figure 33 shows an alternative map that may be used in the region when x is small. du is defined as $\frac{1}{x}$dy in the problem and is used in a manner which prevents an integrand from increasing beyond bounds.



$$\frac{d^2y}{dx^2} = -\frac{1}{x}\frac{dy}{dx} - y$$

Figure 32



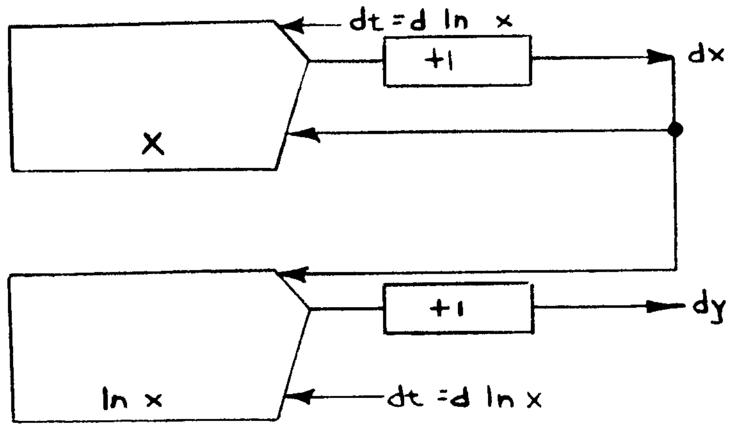
$$\frac{d^2y}{dx^2} = -\frac{1}{x}\frac{dy}{dx} - y$$

Figure 33

A simple map may often be found for a problem by performing a substitution of variables. For example, in the equation,

$$\frac{dy}{dx} = \ln x,$$

the machine independent variable may be used for $d\ln x$ as shown in Figure 34.



$$\frac{dy}{dx} = \ln x$$

$$y = x(\ln x - 1)$$

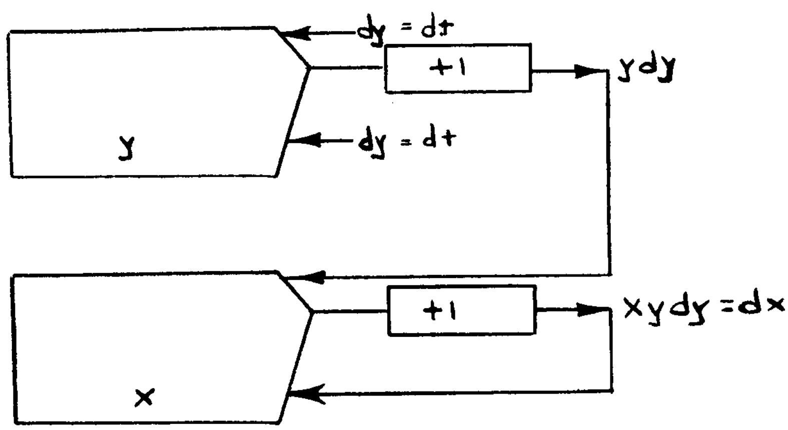
Figure 34

Note that machine time, dt, is not always used as an independent variable in the problem. Machine time may even be used as the problem dependent variable as shown in Figure 35.

The map shown in Figure 35 can be used with the initial condition, x = 1, y = 0, even though

$$\left(\frac{dy}{dx}\right)_0$$

is infinite.



$$\frac{dy}{dx} = \frac{1}{xy}$$

Figure 35

Multiplication of Variables

In most cases the product required can be placed in differential form; products may then be built up using just one integrator per multiplication. For example, the term uvywdx may be generated with four integrators as shown in Figure 36.

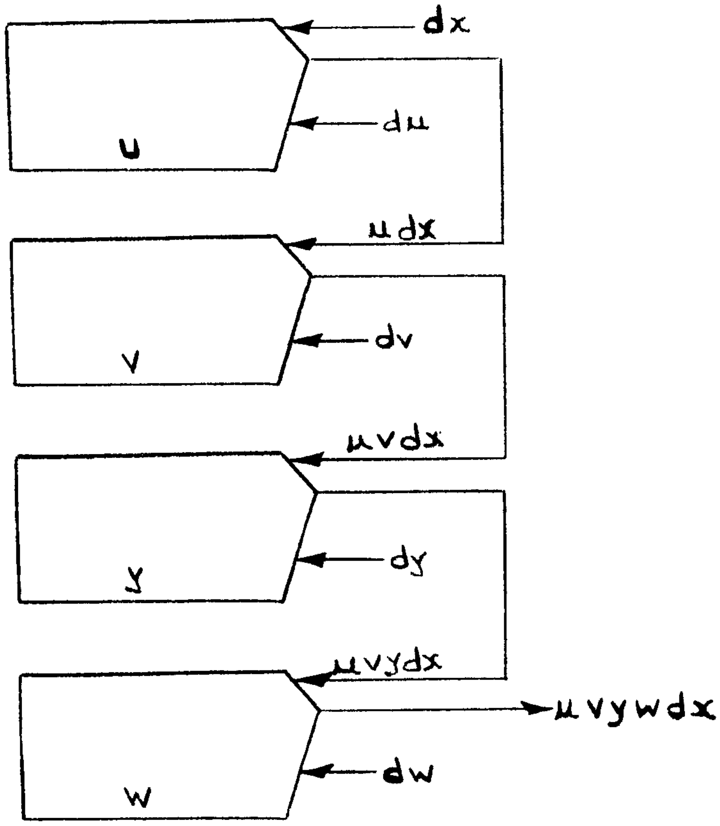


Figure 36

In cases in which the product cannot be placed in differential form, use is made of the relation

$$uv = \int v\,du + \int u\,dv$$

and the corresponding map is shown in Figure 37.

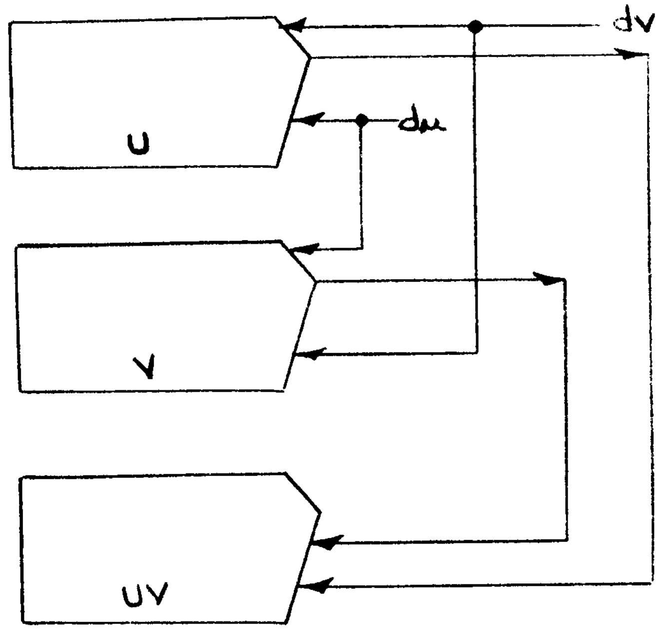


Figure 37

In Figure 1 at the front of this manual, a plot is shown of an elliptical spiral. The picture shows $\frac{dx}{dt}$ plotted against x in the equation

$$\frac{d^2x}{dt^2} + c\frac{dx}{dt} + \omega^2 x = 0$$

The subsequent map is shown in Figure 38.

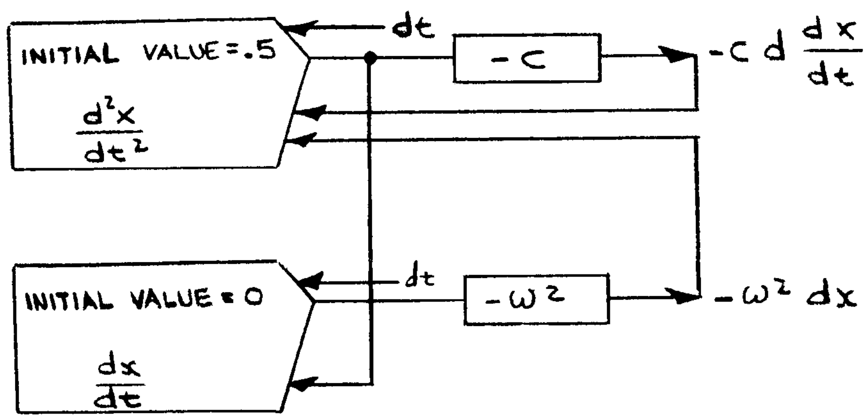


Figure 38

PART V - HOW TO SCALE NUMERICAL VALUES

There are two additional steps to the preparation of a program for the DA-1 after the flow diagram has been prepared. They are the selection of scaling exponents for each integrator and the coding (putting in computer language) of the information from the scaled flow diagram. Scaling will be discussed in this section.

Choice of Scale Powers

Three binary scale exponents must be specified for every integrator. One, S_y, is determined by the maximum value expected to be reached by the integrand, y; one, S_k, is determined by the value of the constant multiplier, K; the third, S_{dy}, specifies the magnitude of variation in y to be represented by a dy increment.

Consider in turn the method of obtaining each:

S_y

To find S_y, the largest magnitude expected to be reached by the integrand is estimated. S_y is the power of 2 such that 2^{S_y} is larger than this maximum. (If in computation the integrand tries to reach or exceed 2^{S_y}, the DA-1 will automatically halt and the overflow neon on the main computer will light.)

> Example: The y variable is the speed of an automobile which is known to remain less than 100 miles per hour.
>
> Since $2^6 = 64$ and $2^7 = 128$, the smallest scale exponent that could be used for S_y is 7.

S_k

S_k is the power of 2 such that 2^{S_k} is larger than the constant multiplier, K.

> Example: If K = 296, find S_k. Since $2^8 = 256$ and $2^9 = 512$, S_k is set equal to 9.

S_{dy}

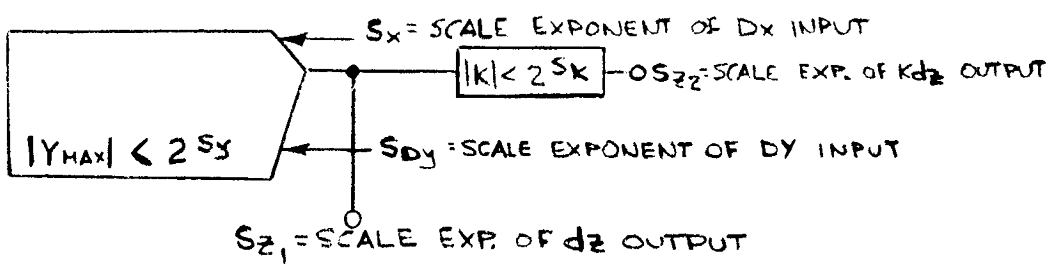
S_{dy} represents the increment of the Y variable. It specifies the value of a single dy input, the smallest variation to be permitted in the integrand. S_{dy} is chosen so that $2^{S_{dy}}$ is the nearest power of 2 to the desired smallest variation.

> Example: In the automobile problem above, the speed is required to the nearest tenth of a mile or better. One-tenth is between one-eighth and one-sixteenth. Since 1/16 mph = 2^{-4} mph, set S_{dy} to be equal to -4.

If an integrator is used as a constant multiplier, make $S_{dy} = S_y - 26$.

All exponents in DA-1 programming are expressed in excess-fifty form. In the preceding three examples, therefore, S_y would be written as 57, S_k as 59 and S_{dy} as 46.

Relationship Between Scaling Exponents



$$S_{z_1} = S_y + S_x$$

$$S_{z2} = S_k + S_y + S_x$$

Figure 39

The first step in scaling a problem is to estimate S_y, S_{dy} and S_k for each integrator-multiplier pair. It is then necessary to determine the compatibility of the scale exponents estimated for each integrator with those of the other integrators. To do so, the additional scaling exponents listed in Figure 39 must be calculated. The scaling exponent which exists at the output of an integrator (S_{z1} or S_{z2}) must be equal to the S_{dy} or S_x for the integrator to which it goes as an input. S_{z1} and S_{z2} are determined by the formulas in Figure 39.

It may prove necessary to modify some of the estimated scaling exponents in order to obtain compatibility.

An example of the procedure for the adjustment of scaling exponents is shown in Figure 40. In Integrator B, S_{dy} was first estimated to be -4; however, the calculation of S_{z2} for Integrator A necessitated that S_{dy} for Integrator B be changed to -6.

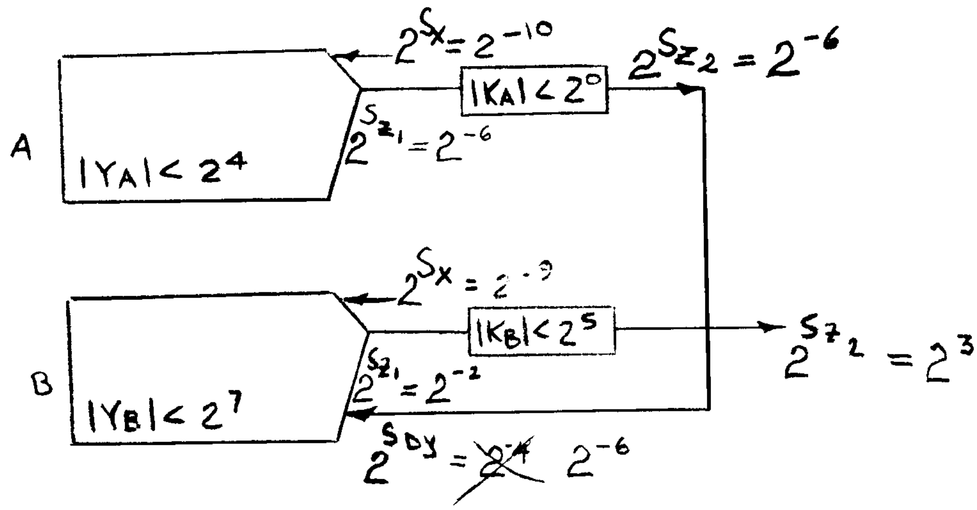


Figure 40

Figures 41 and 42 show the same flow diagram as Figure 40 with the necessary integrator scaling adjustment made in different manners. In Figure 41 the scaling exponent of Y_a in Integrator A was the one changed to obtain compatibility. In Figure 42 the scaling exponent of K_a of Integrator A was changed to accomplish the same result.

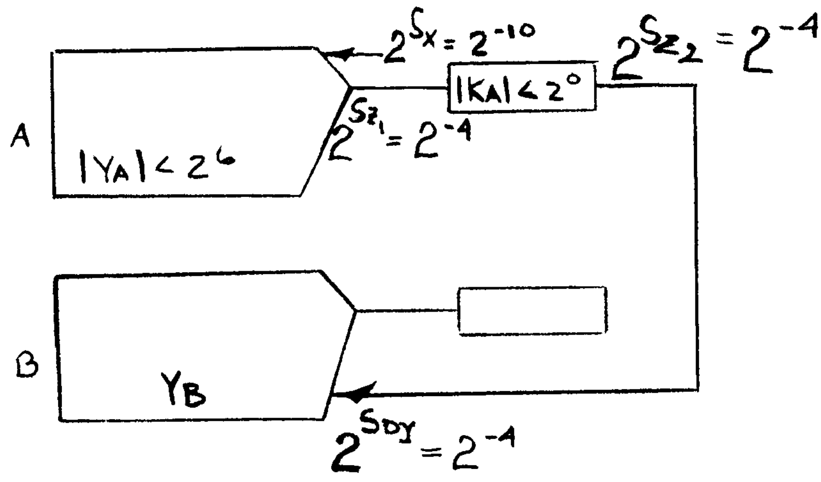


Figure 41

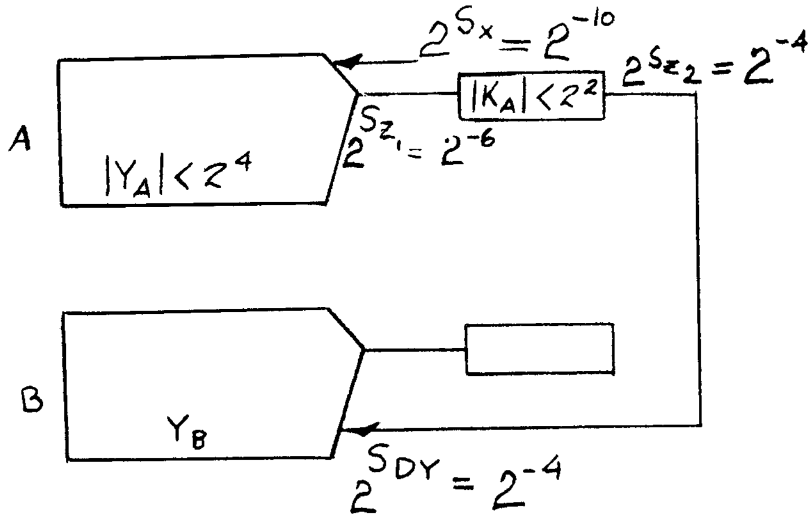
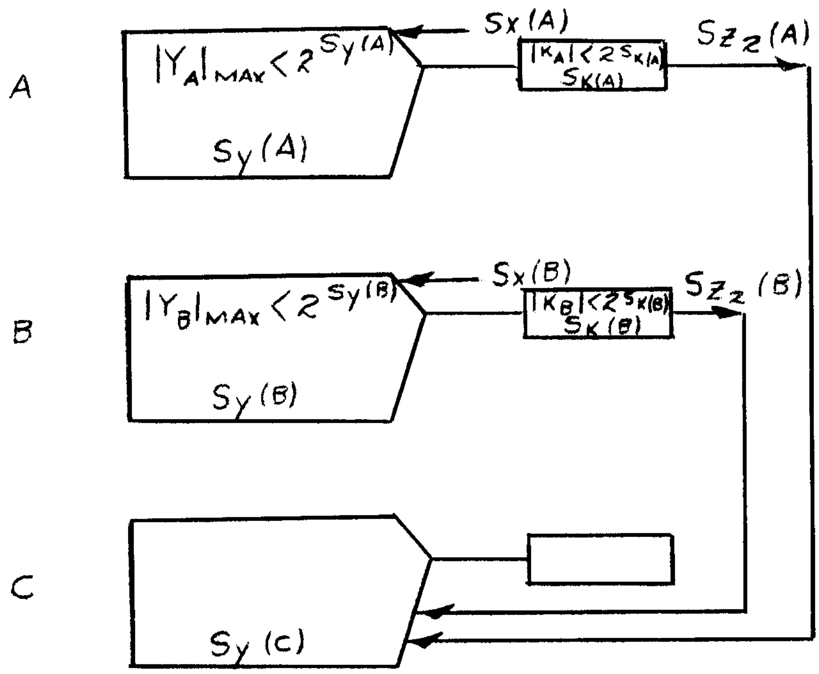


Figure 42

When several integrators provide dy inputs to another integrator, the output scaling exponents of each of the sources must be equal to S_{dy} for the destination integrator.

The scaling exponent of the machine independent variable, dt, may be arbitrarily chosen in order to make the other scaling exponents realizable and practical. A convenient preliminary value for the scaling exponent of dt is the average of the values of $S_{dy} - S_y$ for the integrators in the program.

The scaling relationships described are summarized in Figure 43.



S_{Dy} FOR INTEGRATOR C $= S_{z_2}(A) = S_{z_2}(B)$

$$S_{z_2(A)} = S_{x(A)} + S_{y(A)} + S_{K(A)}$$

$$S_{z_2(B)} = S_{x(B)} + S_{y(B)} + S_{K(B)}$$

Note: These scaling relationships do not apply to integrators coded for decision element operation.

Figure 43

PART VI - HOW TO CODE AND OBTAIN NUMERICAL SOLUTIONS

After the problem has been mapped by means of a flow diagram, after an initial value for y and a value for k has been chosen for each integrator, and after scaling has been completed, the problem may be coded. In coding the scaled flow diagram is transcribed into a set of numerical codes understandable by the computer.

Rules and considerations for the assignment of integrator numbers and for other aspects of coding are given in Table I. Each integrator in the problem is coded by use of the form in Table II. Read-out instructions to the computer are coded in the form given in Table III. The information from the coding forms may then be entered into the computer by following the procedure in Table IV.

If desired, the program may be modified after being filled into the computer. Operation codes with which to do so and other codes for instruction of the DA-1 are listed in Table V.

Since it is quite easy to violate one of the coding rules and procedures while familiarity is being gained with them, facilities are included for the automatic detection of errors.

An error during typein of the program will cause an immediate typeout of a number indicative of the type of error. In Table VI the possible errors and their indicator numbers are tabulated.

TABLE I - Coding Considerations and Rules

1. Integrator 00 automatically has a dx input every iteration. The output of Integrator 00 (dt) is the independent variable that has been called "machine time".

2. If Integrator 00 receives a programmed dx input, positive or negative, computation in the DA-1 will automatically halt. Therefore, the output of an integrator may be used to halt computation by coding the integrator to be the dx input to Integrator 00.

3. An integrator may be coded to receive a dx input from one other integrator.

4. An integrator may be coded to receive dy inputs from any number of integrators. The total value of dy inputs in any one iteration must lie between -7 and +7 increments. Therefore, if information is not known about the total value of dy inputs to an integrator during an iteration, no more than seven integrators should be coded for dy input.

5. The kdz output of an integrator (see Figure 44) may be sent to the input line of any integrator and any number of integrators. The kdz output sent to the integrators, numbered one greater and two greater than the integrator providing the output, will be one iteration old.

6. The dz output of an integrator (see Figure 44) may be sent to the integrators numbered one greater, two greater, three greater, and four greater than the integrator providing the output.

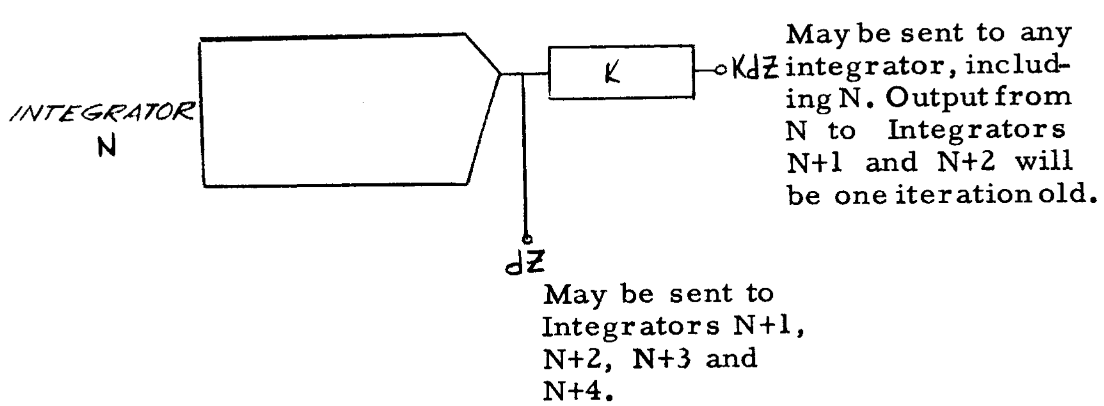


Figure 44

7. Numerical values for y and K are expressed in floating decimal point form. The numerical portion is written as decimal point followed by seven digits. The exponent portion may range between 10^{-4} and 10^{+9}. The exponent of 10 is expressed as an excess-fifty number preceding the numerical portion.

   Example: The number "4963" would be expressed in computer language as "54 4963000"

8. The DA-1 may be instructed to operate as a 27 integrator machine, a 54 integrator machine, an 81 integrator machine, or a 108 integrator machine. The integrator numbers which may be used in each case are:

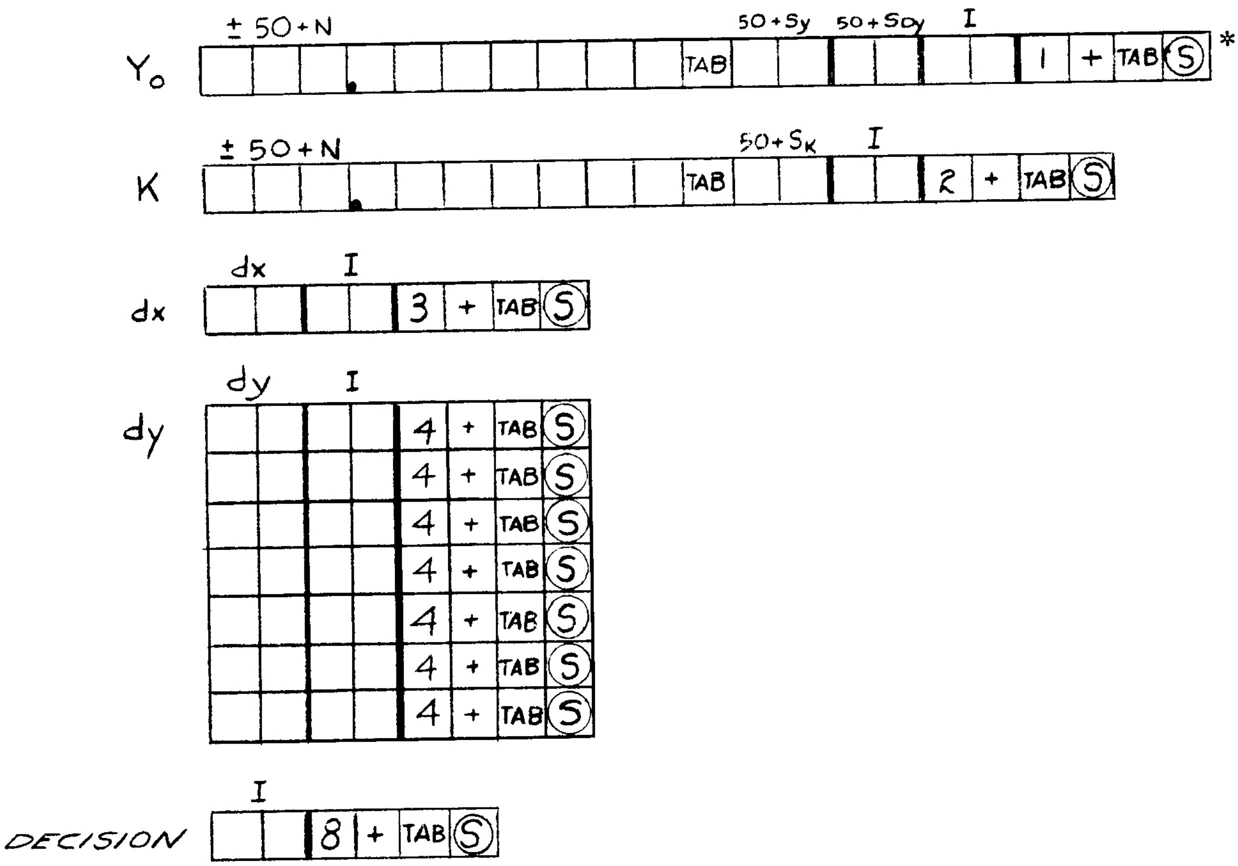
27 integrator operation:	Integrators 0, 4, 8, 12 104
54 integrator operation:	Integrators listed above, plus 1, 5, 9, 13 105
81 integrator operation:	Integrators listed above, plus 2, 6, 10, 14 106
108 integrator operation:	Integrators listed above, plus 3, 7, 11, 15 107

9. The numerical value of the difference between the scaling exponents S_y and S_{dy} (that is, $S_y - S_{dy}$) must be between 3 and 26 inclusive.

10. The decimal exponent of an integrand coded for typeout must be large enough to permit typing the largest possible integrand value as a fraction.

    Example: If the binary scale exponent, S_y, equals 10, the integrand could become larger than 1000 but will remain under 10,000. Therefore the smallest permissible value of the decimal exponent of the integrand would be +4. Larger values could be used.

TABLE II - Coding Form



Y_0 and K are expressed as sign, decimal exponent in excess-fifty form, and decimal fraction. The number "+8263", for example, would be written "+54.8263000".

"I" is the number of the integrator being filled.

"dx" is the number of the integrator from which dx comes.

"dy" is the number of an integrator from which dy comes. Though space for only seven dy inputs are shown, more may be filled if desired.

The "Decision" line should not be typed unless the integrator is to be a decision integrator.

Integrator numbers are always written with two digits. Integrator numbers 100 to 107 are written as u0 to u7.

> *The earlier model typewriter has no Ⓢ key; wherever the Ⓢ key is indicated, use the "s" key on the alphabetic keyboard.

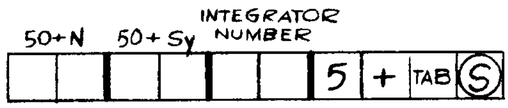
TABLE II - Coding Form (cont.)

A dx or dy input specified as 00 to 99 or as u0 to u7, refers to the multiplied (kdz) output of the integrator from which the input comes. To specify that the input to any integrator, N, be from an unmultiplied (dz) integrator output, write the integrator number of the dx or dy input as follows:

Write z1 if input from Integrator N-1
Write z2 if input from Integrator N-2
Write z3 if input from Integrator N-3
Write z4 if input from Integrator N-4.

TABLE III - Readout Control

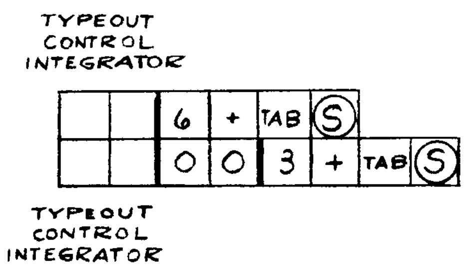
1. To list an integrator for typeout



The contents of the Y registers of any twelve, or fewer, integrators may be automatically typed out at selected intervals.

Typeout is decimal, and is expressed in floating point form as a decimal fraction.

2. To code typeout interval



Any integrator may be selected to be typeout control integrator. Integrands listed for typeout will be typed out whenever the typeout control integrator has a dz output.

Explanation: As stated in Table IV, computation begins with the command "x+(TAB)(S)". This command causes typeout followed by restart of computation when the DA-1 is halted. In the first of the two coding steps above, the R register of the typeout control integrator is set to zero (rather than .5), so that all typeout intervals will be equal; in the second, Integrator 00 is coded to have the typeout control integrator as its dx input. This second step causes computation to halt when the typeout control integrator has an output.

TABLE IV - Operating Instructions

1. Initially, the three switches on the typewriter base ("Compute", "Enable" and "Punch") should be off.

2. Insert magazine DAPPER-1A in rewound position on the photoreader.

3. Plug in Accessory DA-1 (two plugs). Turn on power switch on back panel of DA-1.

4. Set switches on back panel of DA-1 as follows:

   Switches 9 and 13 on, others off, for 27 integrator operation.
   Switches 8, 9, 12 and 13 on, others off, for 54 integrator operation.
   Switches 7, 8, 9, 11, 12 and 13 on, others off, for 81 integrator operation.
   All switches on for 108 integrator operation.

5. Turn on G-15 Start Switch. Wait several minutes for warm-up cycle to be completed.

6. Hold RESET button in until the DC voltage is turned on.

   Two blocks of tape will be read in. The first block is the Number Track for the G-15 Computer. The second block is a loading and memory - setup routine. Initial values of zero are put in Y registers and values of +1 are put in K registers.

7. Wait until the green "Ready" light turns on. Then put Compute switch on typewriter base to GO.

   The typewriter will execute a carriage return. Neons on G-15 power control panel will be in TYPEIN configuration.

8. Type "N TAB Ⓢ" where N refers to the number of integrators in one iteration according to the following code:

   N=1 for 27 integrator operation
   N=2 for 54 integrator operation
   N=3 for 81 integrator operation
   N=4 for 108 integrator operation

   The computer will type out the number of integrators and additional tape will be read into the computer. When tape reading has been completed, the typewriter will execute a carriage return. Neons on the G-15 power control panel will be in the TYPEIN configuration.

TABLE IV - Operating Instructions (cont.)

9. The coded program may now be typed directly from the coding forms of Tables II and III. All contents of the forms which are in blocks must be typed. The remaining information is merely explanatory.

   Input information is automatically converted into proper binary form and stored on the memory drum. Initial values of integrands are stored in Initial Conditions Registers on the drum. After the filling of integrand or constant multiplier values, the computer will retype the number in hexadecimal notation. After any fill operation, the computer will retype the operation code.

10. Type "y+(TAB)ⓢ"

    This instruction inserts initial values of all integrands from the Initial Conditions Registers into the Y registers.

11. For reference during readout, the integrator numbers and exponents of integrators to be read out may be automatically typed by the computer.

    Type "u-(TAB)ⓢ"

    Integrator numbers of integrators to be read out will be typed out.

    Type "v-(TAB)ⓢ"

    Binary scale exponents of read-out integrators will be typed out.

    Type "v+(TAB)ⓢ"

    Decimal exponents of read-out integrators will be typed out.

12. Type "x+(TAB)ⓢ"

    The computer will type out initial values of integrands and will then start computing, typing out computed results in decimal form at the programmed intervals.

<u>Halt Control</u>

Computation will halt automatically either when Integrator 00 receives a programmed dx input or an integrand overflow occurs. If it is desired to halt computation at other times, and thus permit typein of further instructions:

1. Move Compute switch to BP when control-panel neons are in the TEST DA-1 OFF or TEST READY configurations.

   The DA-1 will halt after the next typeout.

2. After the typeout is completed, move the Compute switch to OFF.

3. With the Enable switch on, type "f".

4. Move the Compute switch to GO.

   After one more typeout, the control-panel neons will revert to the TYPEIN configuration.

To halt computation in case of program difficulties:

1. With the Compute switch OFF and the Enable switch on, type "Ⓢcf".

2. Move the Compute switch to GO.

   The control-panel neons will be in the TYPEIN configuration.

Memory Punchout

The contents of DA-1 memory can be punched on tape. A problem in memory may be punched out at any stage of computation and later reloaded. Computation may then proceed either from the beginning of the problem or from the point where it was interrupted. If computation is to be resumed exactly from the point at which it was interrupted, dx and dy inputs should not be addressed to integrators 00, 01, 02 and 03.

The procedure for initiating memory punchout is:

When the neons on the power control panel are in the TYPEIN configuration (follow Steps 1 - 4 under Halt Control, above, to obtain this configuration), type "9 +(TAB)Ⓢ".

The contents of DA-1 memory will be punched on tape. A bell will ring after about ten minutes to indicate the last block of tape being punched.

Notes: Initial conditions are automatically re-inserted in the integrators after punching is completed. Therefore, the problem is in its original state in the memory of the computer.

Reloading Punched Out Contents of Memory

The contents of DA-1 memory that were punched on tape can be re-entered into the computer. After DAPPER-1A has been read in, the procedure is:

1. Move the Compute switch to OFF and then remove Magazine DAPPER-1A without rewinding.

2. Insert the tape containing the punched out contents of memory on the photo-reader.

3. With the Enable switch on, type "p".

   Wait for the photo-reader light to go off.

   If the bell rings and the computer halts, a reading error has occurred. In this case, rewind the tape and repeat Step 3.

4. Move the Compute switch to GO.

   The subsequent blocks on the tape are read in. In case of error the tape is automatically reversed, the bell rings, and the block is re-read.

TABLE IV - Operating Instructions (cont.)

5. Check the switch settings on the back panel of the DA-1 to be sure the switches are in the same positions they were before the memory punchout of the problem. (See TABLE IV - Operating Instructions, Step 4, page 55.)

6. Replace Magazine DAPPER-1A on the photo-reader.

   Input information may be typed under control of the DAPPER-1A routine.

7. To insert initial conditions, type "y(TAB)Ⓢ".

8. To resume DA-1 computation, from the point where memory was punched out, omit Step 7 and type "x(TAB)Ⓢ".

<u>Monitor Operation</u>

Programs may be monitored if desired. In this operation the contents of readout integrators are automatically typed out every iteration. The procedure is:

1. Move Compute switch to GO.

2. Type "u +(TAB)Ⓢ".

   The current values of readout integrands will be typed. On completion of the typeout, the control-panel neons will revert to the TYPEIN configuration.

3. Type "x -(TAB)Ⓢ".

   The program will now be executed and readout integrands will be typed out each iteration.

To stop monitoring, the procedure is:

1. Move the Compute switch to BP when the control-panel neons are in the TEST READY configuration.

   The computer will execute one more typeout and then halt.

TABLE IV - Operating Instructions (cont.)

2. When the computer has halted, move Compute switch to OFF.
3. With the Enable switch on, type "Ⓢcf".

   The neons will revert to the TYPEIN configuration.

To skip typeout intervals while monitoring:

In order to monitor program execution at points away from the start of computation, move Compute switch to BP and type "x-(TAB)Ⓢ". Moving the Compute switch to OFF and back to BP will alternately start and stop the DA-1. At each halt, readout integrands will be typed out. Normal monitor operation can be resumed by moving the Compute switch to GO.

Caution: Move the Compute switch only when the Halt neon is lit or control-panel neons are in the TEST READY configuration.

Overflow Detection

After "x+(TAB)Ⓢ" has been typed:

If an integrand overflows during normal computation, the DA-1 will halt. The photo-reader will read in a section of tape that contains overflow instructions. The code digits "zz" will be typed out to indicate that the halt was due to an overflow. The integrand which overflowed will be typed out in hexadecimal notation. The integrator number in which the overflow occurred will be typed out in decimal notation. The integrand which overflowed will be cleared to zero. The current values of all readout integrands will be typed out.

After "x-(TAB)Ⓢ" has been typed:

If an integrand overflows during monitor operation, the DA-1 will halt and the Overflow neon will light. However, in order for the overflow detection and typeout operations listed above to occur, "7+(TAB)Ⓢ" must now be typed.

TABLE V - Operation Codes for Program Control

Code	Operation
I 3 - TAB (S)	Remove dx input from Integrator I.
I 4 - TAB (S)	Remove dy inputs from Integrator I.
I 8 - TAB (S)	Change Integrator I from decision use to normal use.
5 - TAB (S)	Remove list of integrators coded for readout.
7 + TAB (S)	Type out integrand (hexadecimal notation) and integrator number (decimal notation) which caused overflow. Clear overflow integrand to zero. Type out integrands of readout integrators. Turn off overflow neon.
9 + TAB (S)	Punch out contents of DA-1 memory on tape.
U + TAB (S)	Type out decimal portions of readout integrands.
U - TAB (S)	Type out integrator numbers of readout integrators.
V + TAB (S)	Type out tens exponent of readout integrands in excess-fifty form.
V - TAB (S)	Type out S_y of readout integrands in excess-fifty form.
W + TAB (S)	Recall DAPPER-1A loading routine.
X + TAB (S)	Type out decimal portions of readout integrands and compute, typing out at programmed intervals.
X - TAB (S)	Compute, typing out decimal portions of readout integrands every iteration.
Y + TAB (S)	Insert initial values of integrands in all integrators.

TABLE VI - Error Indicators During Typein

Typed-Out Indicator	Operation in Error	Type of Error
59	1+ Fill Y_0 2+ Fill K_0	(50+N) exceeds limits
58 or 57	1+ Fill Y_0 2+ Fill K_0	S_y too large or too small or scaling in error
56	1+ Fill Y_0	S_y - S_{dy} beyond limits
53	dx or dy (not coded 1+, 2+, 3+ or 4+)	Integrator number greater than 107
54 or 53	3+ dx input 4+ dy input	Integrator number of source not legitimate number.
52	3+ dx input 4+ dy input	Integrator number of destination greater than 107
51	1+ Fill Y_0 2+ Fill K_0	Integrator number greater than 108
49	3+ dx input 4+ dy input	Machine error
zzzzzzz	u+ x+ x-	Overflow in computation preceding readout
z000000 or 000000z	u+ x+ z-	Sy in readout list is too large or too small
No typeout after "u (TAB) Ⓢ" or "x (TAB) Ⓢ"	u+ Typeout x+ Typeout z+ Typeout	(50+N) in readout list exceeds limits

TABLE VI - Error Indicators During Typein (cont.)

The decimal exponent of a readout integrand must be large enough to permit typing the largest possible integrand value as a fraction. (See Table I.) If 50+N is too small or much too large, "z000000" or "000000z" would be the typed out error indicator.

If an error indicator is typed out accompanied by <u>two</u> carriage returns, rather than one, the type of error is not determined by the indicator number. The error is one in which a zero preceded "(TAB)Ⓢ". Zero in that position corresponds to no legitimate operation code.

TABLE VII - Graphical Output

Facilities are included in the DA-1 for the use of graph-plotter output.

A graph-plotter accessory is available which receives Δy and Δx increments from the DA-1 and plots them in 0.01 inch increments on a standard roll of paper, 1 foot by 100 feet.

Programming specifications for the use of the plotter are summarized below.

<u>Output to Graph-Plotter Plugged Into Graph-Plotter Jack</u>

Δy The output of any integrator in the series:

1, 5, 9, 13, 17, ... 105

may provide the Δy output to the plotter. The integrator will provide Δy if it is coded to be the dx input into Integrator 03.

Δx The output of any integrator of the two series:

0, 4, 8, 12, 16, ... 104
1, 5, 9, 13, 17, ... 105

may provide the Δx output to the plotter. The integrator will provide Δx if it is coded to be the dx input into Integrator 02.

<u>Output to Graph-Plotter Plugged Into Graph-Plotter-Follower Jack</u>

Δy The output of any integrator may provide the Δy output, except integrators in the series:

2, 6, 10, 14, 18, ... 106.

The integrator will provide Δy if it is coded to be the dx input into Integrator 01.

Δx The unmultiplied output of Integrator 107 will provide the Δx output to the plotter. No additional coding is necessary.

TABLE VIII - Control Panel Neon Configurations

Significance	Pattern	
TYPEIN	○ ○	Characteristic
	○ ○ ○	Command Line
	● ● ● ○ ○	Source
	● ● ● ● ●	Destination
	○ ● ● ○ ○	Input-Output
TEST READY	○ ○	Characteristic
	● ● ● ○ ○	Source
	● ● ● ● ●	Destination
TEST DA-1 OFF	● ●	Characteristic
	● ○ ○	Command Line
	● ● ● ○ ○	Source
	● ● ● ● ●	Destination
TEST PUNCH SWITCH	○ ●	Characteristic
	● ○ ○ ○ ●	Source
	● ● ● ● ●	Destination
START DA-1	○ ○	Characteristic
	● ○ ○	Command Line
	● ○ ○ ● ●	Source
	● ● ● ● ●	Destination
TYPEOUT	○ ○	Characteristic
	● ○ ○	Command Line
	● ○ ○ ● ●	Source
	● ● ● ● ●	Destination
	○ ● ○ ○ ●	Input-Output

"Typein", "Test Ready" and "Test DA-1 Off" configurations occur at the times specified in Table IV.

The "Test Punch Switch" pattern means that the Punch switch on the typewriter base is on. The switch should be turned off for Routine DAPPER-1A.

The "Start DA-1" or the "Typeout" pattern will occur when the Compute switch is in the BP position after "x-(TAB)ⓢ".

The "Typeout" configuration will occur when the Compute switch is in the BP position after "x+(TAB)ⓢ".

POINTS TO REMEMBER

1. The value in the Y register of an integrator coded as a servo, or as an adder, must remain close to zero throughout computation for results to be accurate.

2. If the value in a K register is to be +1, that value need not be inserted by the programmer. Routine DAPPER-1A puts +1 in all K registers unless some other value is typed in.

3. If a mistake occurs during the typein of dx or dy inputs, remove the inputs in error by using the proper operation code from Table V before retyping the correct input information.

4. In turning on the G-15 and DA-1 combination, be sure that the warm-up cycle in both units is completed before turning on DC voltage.

5. If the computer is used when Accessory DA-1 is not connected, the dummy plug must be in place in back of the computer. Whenever the computer is in use and Accessory DA-1 is connected, the Power switch in the DA-1 must be on.

   Computer power must be off while the DA-1 is being attached or unattached.

6. To halt and return to the "Permit Typein" state in case of program difficulties, type "Ⓢcf" with the Enable switch on and the Compute switch off.

7. Routine DAPPER-1A puts initial values of zero in the Y Registers. There can be no output from Integrator 00 unless it is programmed. Accordingly, it is necessary to enter a constant, normally + 1, in the integrand of Integrator 00 so that an output may occur.

APPENDIX

The Program Preparation Routine, DAPPER-1A

The scaling and coding instructions of Parts V and VI are based on the use of the phototape magazine DAPPER-1A, Differential Analyzer Program Preparation and Execution Routine, Number 1A. This routine has been designed for typewriter input and typewriter or graph-plotter output. A different program preparation routine would provide different input-output characteristics.

Internal Information Storage With Accessory DA-1

Accessory DA-1 utilizes the memory drum of the general purpose computer for information storage during computation. Each of the 108 word times on the drum periphery corresponds to one integrator time. It is necessary to know the distribution and nature of the information stored on the drum in order to code the DA-1 without the use of a program preparation routine, or in order to write a program which simultaneously uses the differential analyzer and general computation facilities of the G-15 system.

Form of Information

Numerical information for the DA-1 is arranged differently from numerical information for general computation.

The basic G-15 word consists of a sign bit, in bit position 1 and twenty-eight bits of magnitude, in bit positions 2 through 29. The DA-1 word consists of twenty-seven bits of magnitude, in bit positions 1 through 27, and a sign bit in bit position 28; bit position 29 is blank. (The presence of a "one" in bit position 29 of line 17 during computation indicates the overflow of the register represented by the word.)

In the DA-1 word, a negative number is written in complemented form and a negative sign is represented by a "one" bit in bit position 28, the sign position.

Memory Drum Utilization

There are twenty 108-word channels or "long lines" on the magnetic drum in the G-15 Computer. When the DA-1 is attached, these channels are used as follows:

	Line
Control Information	18
Y Registers	17
R Registers	16
K Registers	15
K_r Registers	14
DY Addresses (If DAPPER-lA is used)	
27 Integrator Operation	13
54 Integrator Operation	12, 13
81 Integrator Operation	11, 12, 13
108 Integrator Operation	10, 11, 12, 13
DX Addresses (If DAPPER-lA is used)	
27 Integrator Operation	9
54 Integrator Operation	8, 9
81 Integrator Operation	7, 8, 9
108 Integrator Operation	6, 7, 8, 9
Y_0 Registers (If DAPPER-lA is used)	
27, 54 or 81 Integrator Operation	10
108 Integrator Operation	5

Routine DAPPER-lA requires for its program the remaining long lines on the drum. Long memory lines available for general use by the programmer when the DA-1 is in operation are summarized below.

27 Integrator Operation	5, 6, 7, 8, 11, 12
54 Integrator Operation	5, 6, 7, 11
81 Integrator Operation	5, 6

Line 18

Line 18 contains control pulses which are automatically filled by Routine DAPPER-lA. The filling of these pulses must be programmed if DAPPER-lA is not used. Every word (one word time corresponds to one integrator time) contains either two or three control pulses. A "one" must be inserted in bit position 26 of every word in the line. If an integrator is to be used as a decision integrator, a "one" must be placed in bit position 28 of the correspondingly-numbered word in Line 18. In each word that corresponds to an integrator used in the program, a "one" must be placed in the bit position, relative to the binary point, the number of which equals S_{dy} for the integrator. If the integrator is one not used in the program, a "one" is placed in bit position 5.

> Example: The binary point of an integrand is between bit positions 15 and 16. If S_{dy} = +4, a one is placed in bit position 19 of the correspondingly-numbered word in line 18. If S_{dy} = -5, a one would be placed in bit position 10 instead.

Line 17

Line 17 holds the Y registers or integrands. Bit position 29 of every word is blank; bit position 28 holds the sign; with routine DAPPER-1A, digit positions 2 through 27 hold the magnitude in binary form. The location of the binary point in the integrand is determined by the scaling exponent S_y. S_y is equal to the number of binary positions (not including bit positions 28 or 29) to the left of the binary point. A negative number is written in complemented form.

> Example: If y = +1101.1001 and S_y = 6, in an integrator, the correspondingly-numbered word on line 17 would contain:
>
> 00001101.10010000000000000000000
>
> BIT POSITION 29 — SIX BINARY PLACES — BIT POSITION 1
>
> If y = -1101.1001 and S_y = 6, in an integrator, the correspondingly-numbered word on line 17 would contain:
>
> 01110010.01110000000000000000000
>
> BIT POSITION 29 — SIGN — BIT POSITION 1

Routine DAPPER-1A automatically performs the conversion from the decimal input program to the form which has been described.

Line 16

Line 16 holds the R registers. As stated in the text, before DA-1 computation is begun, a value of one-half in DA-1 language* should be inserted in the R register of every integrator used in the program except the typeout control integrator. With Routine DAPPER-the insertion is automatic when the operation code "y(TAB)Ⓢ" is typed.

Lines 15 and 14

Lines 15 and 14 hold the K and K_r registers, respectively, in forms which are exactly like the forms of Y and R in Lines 17 and 16. A value of one-half should be inserted in the K_r registers of integrators used in the program.

The DA-1 uses "short lines" 21 and 22 on the memory drum for integrator output storage. In combined operation of the DA-1 and G-15 lines 21 and 22 are not available when the DA-1 is on; if the contents of these lines are disturbed when the DA-1 is off, computation can not be resumed at the same point.

Dy and Dx Addresses

The location and insertion of pulses on the drum to signify the dy and dx addressing of each integrator is best done automatically by DAPPER-lA or some similar routine. If it is desired to insert the pulses directly, the required pulse positions on the drum may be located easily with the aid of the DA-1 addressing wheel.

Pulses must be placed in bit position 29 of words N - 1 (mod 4) in the drum channel specified by the wheel. The wheel, a form of circular slide rule, is available on request.

Memory Punchout

Lines 05 through 18 are punched on tape during a memory punchout if a problem is set for 108-integrator operation. If a problem is set for less than 108 integrators, Lines 07 through 18 are punched.

*2000000 in hexadecimal form.

NUMERICAL VALUES FOR PLOT OF ELLIPTICAL SPIRAL

The plot is shown in Figure 1.

The flow diagram is shown in Figure 38.

The numerical values used to obtain this graphical solution are:

$C = .0625$

$\omega = 1$

Initial value of $\frac{d^2x}{dt^2} = .5$

Initial value of $\frac{dx}{dt} = 0$

$S_{dy} = -6$

ΔY to plotter $= .5625\, d\left(\frac{dx}{dt}\right)$

ΔX to plotter $= .9375\, dx$