Copy No. 2

April 1954

OPERATION MANUAL

DIGITAL DIFFERENTIAL ANALYZER

MODEL D-12

BENDIX COMPUTER

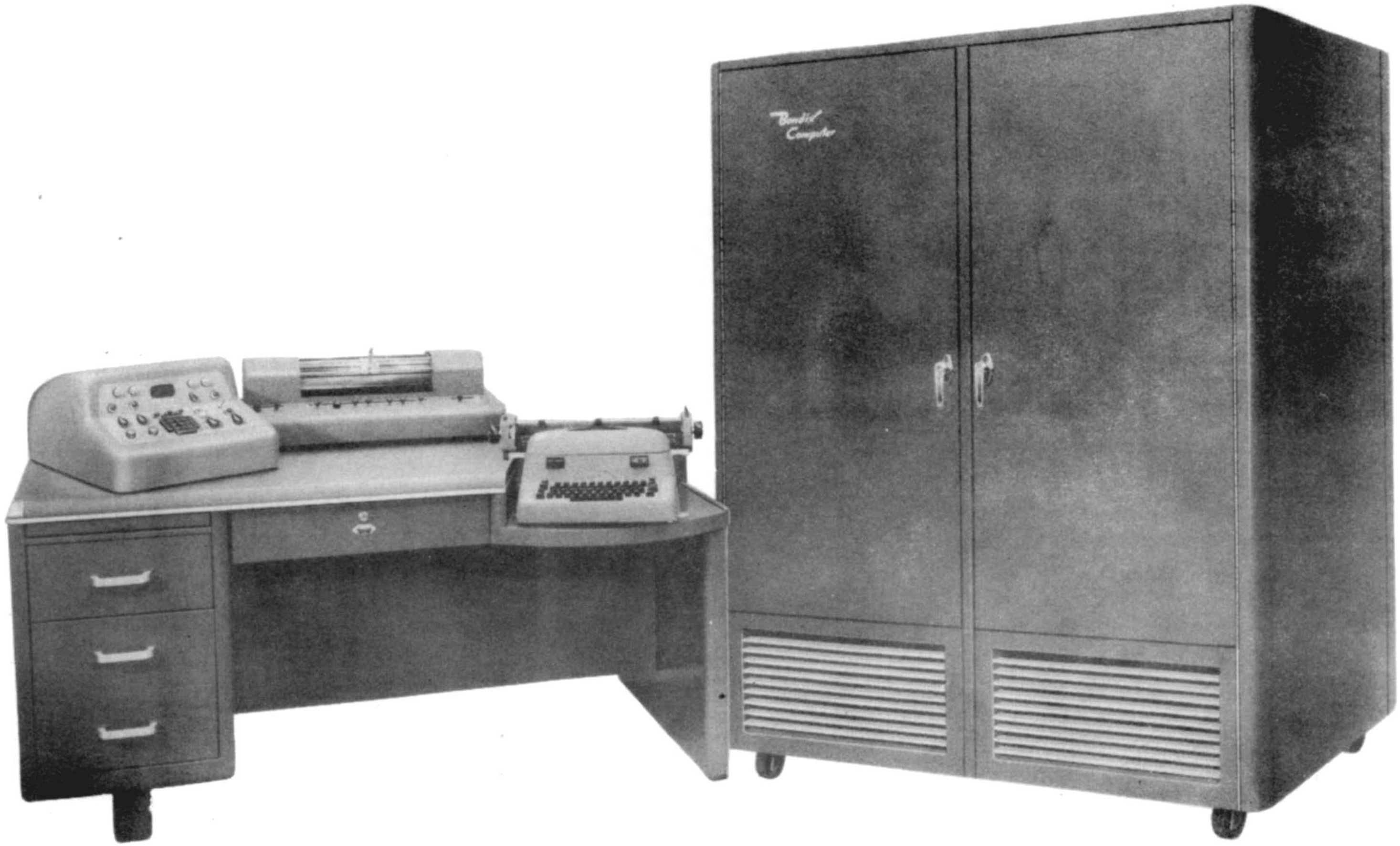
Division of Bendix Aviation Corporation

5630 Arbor Vitae Street, Los Angeles 45, California

CONTENTS

THEORY OF OPERATION

OPERATING PROCEDURES


Bendix Computer

INTRODUCTION

The purpose of this manual is to provide a complete description of the mathematical theory of the Bendix Digital Differential Analyzer, Model D-12, together with a detailed account of the procedures employed in operating the computer. The first section deals with the method of integration, and the mapping, coding and scaling of problems preparatory to computation. The second section indicates the sequence of operations that must be performed in running a problem, including such ancillary procedures as the preparation of tapes and the interpretation of the visual display. The two sections contain, therefore, all the information that is necessary for the personnel who are to operate the computer.

THEORY OF OPERATION

1.1 Integrators

The Bendix Digital Differential Analyzer is an electronic computer that solves ordinary linear or non-linear differential equations by numerical integration. This integration is accomplished by means of a network of interconnected operational entities, called integrators, that sequentially perform simple quadratures in accordance with a fixed program.

Each integrator performs a repeated cycle of operations. In any cycle, say the i-th, the integrator receives a primary incremental input, $(\Delta X)_i$, and a secondary incremental input, $(\Delta Y)_i$, and emits an incremental output, $(\Delta Z)_i$, (see fig. 1).

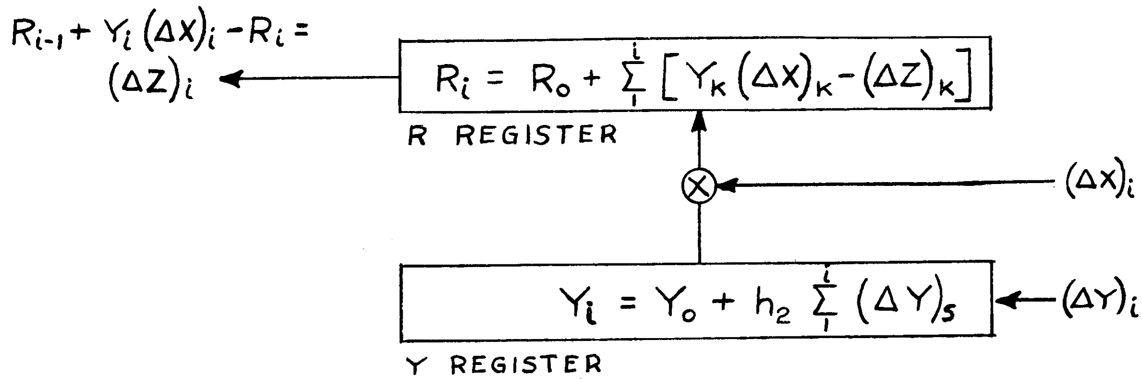


FIGURE 1

The primary incremental input, $(\Delta X)_i$, notifies the integrator of changes in the variable X relative to h_1, a predetermined positive increment in X. If X increases by h_1, $(\Delta X)_i$ is assigned the value +1; if X decreases by h_1, $(\Delta X)_i$ is assigned the value -1; while, if X undergoes a change whose magnitude is less than h_1, $(\Delta X)_i$ is assigned the value 0. The secondary incremental input, $(\Delta Y)_i$, notifies the integrator of similar changes in the variable Y relative to a corresponding predetermined positive increment h_2. The variation in Y is such that its magnitude never exceeds one.* Two arithmetic registers, a Y Register and an R Register, are connected with each integrator. The Y Register accumulates the sum of an initial value, Y_0, and a running total of the secondary inputs multiplied by h_2, to form the value

(Eq. 1) $$Y_i = Y_0 + h_2 \sum_{s=1}^{i} (\Delta Y)_s .$$

The capacities of the Y and R Registers are such that

$$-1 \leq Y_i \leq +1; \qquad 0 \leq R_i < +1.$$

During the i-th cycle, depending upon whether it is positive or negative, or zero, the primary incremental input, $(\Delta X)_i$, causes the contents of the R Register to be increased or decreased by the current total, Y_i, in the Y Register, or to be left unchanged. Consequently, the value of the contents of the R Register will represent a fractional part of an

*Scaling is required when the magnitude of Y exceeds one (see section 1.9.2). The Y Register actually has a larger capacity; namely, $-2 \leq Y_i < +2$, but this added capacity is only used in decision integrators (see section 1.5.3).

increment of area that corresponds to a rectangle of width $h_1 = 1$ and height $+1$, the full height of the diagram in fig. 2a. The program being described is therefore analogous to rectangular integration in which the area under a curve is approximated by summing the areas of a series of rectangles.

Since the R Register is not cleared from one cycle to the next, it will intermittently overflow. This overflow is associated with an incremental change in a variable Z, and constitutes the incremental output, $(\Delta Z)_i$. In any cycle, the R Register can overflow positively, fail to overflow, or overflow negatively. Thus, during the i-th cycle, depending upon whether

$$R_{i-1} + Y_i(\Delta X)_i \geqslant 1, \quad 1 > R_{i-1} + Y_i(\Delta X)_i \geqslant 0, \quad \text{or}$$

$$0 > R_{i-1} + Y_i(\Delta X)_i,$$

the incremental output, $(\Delta Z)_i$, is assigned the value $+1$, 0, or -1.* The value remaining in the R Register at the end of the cycle, $R_i = R_{i-1} + Y_i(\Delta X)_i - (\Delta Z)_i$, represents a fractional part of an increment in Z; i.e., a fractional part of a predetermined positive increment h_3. Accordingly, the output,

*It is possible to cause an integrator to reverse the sign of its output. This is indicated by placing a minus sign in the middle of the right-hand side of the integrator schematic (see fig. 4).

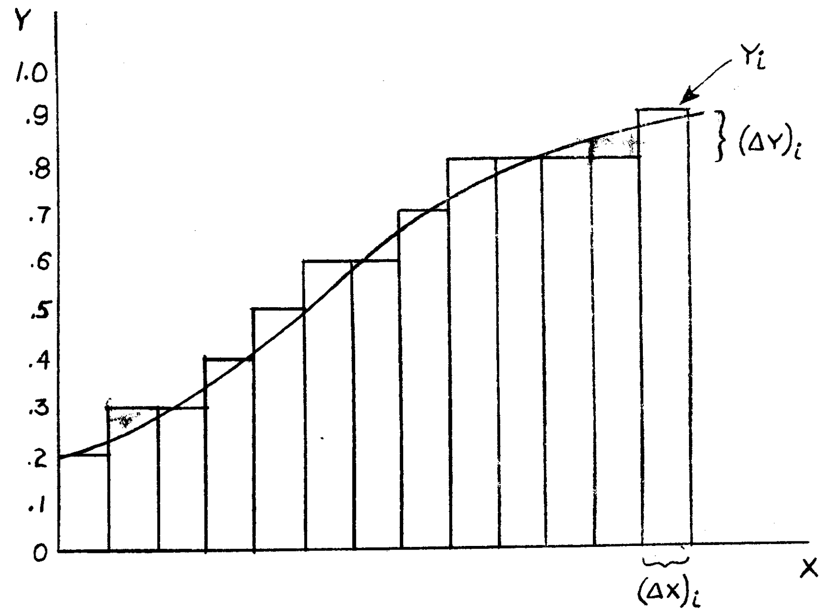


FIGURE 2a

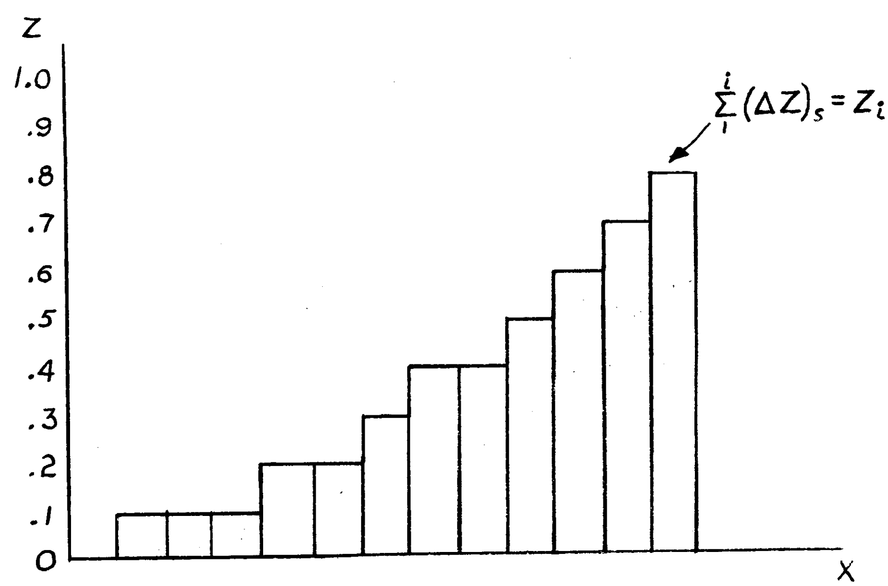


FIGURE 2b

$(\Delta Z)_i$, bears the same relation to the variable Z relative to h_3 as do the inputs $(\Delta X)_i$ and $(\Delta Y)_i$ to the variables X and Y relative to h_1 and h_2.

The primary incremental input, $(\Delta X)_i$, may relate to an increment of time or be the incremental output from any specified integrator. The secondary incremental input, $(\Delta Y)_i$, may be the sum of the incremental outputs from a number of arbitrarily selected integrators. The interconnections of the several integrators depend upon the differential equation being solved, and the X, Y, and Z values associated with each integrator are normally renamed to correspond to variables defined by the problem.

1.2 Round-off

In the Bendix Digital Differential Analyzer the round-off correction is analogous to that used in any digital operation. At the commencement of the first integration cycle, and before the incremental inputs $(\Delta X)_1$ and $(\Delta Y)_1$ have been received, the initial value, R_0, of the quantity in the R Register is arbitrarily set equal to 0.5. Since $0 \leq R_i < 1$, we have $0 \leq 0.5 + r_i < 1$, where r_i is the fractional remainder of an incremental output left in the R Register at the end of the i-th integration cycle. With $-0.5 \leq r_i < 0.5$, the incremental output, $(\Delta Z)_i$, is rounded off to the nearest half increment, $(1/2)h_3$. The numerical approximation to the area under the curve shown in fig. 2a is indicated in fig. 3 as a tabulation of the numerical results that would be obtained in the first 13 integration cycles. In this tabulation, each product, $Y_i(\Delta X)_i$ is added to R_{i-1}, yielding R_i and $(\Delta Z)_i$. In fig. 2b, the values, Z_i, are plotted against X_i.

i	$Y_i(\Delta X)_i$	$(\Delta Z)_i$	R_i	$r_i = R_i - .5$	Z_i
0	------	-----	.5	.0	0
1	.2	0	.7	.2	0
2	.3	+1	.0	-.5	1
3	.3	0	.3	-.2	1
4	.4	0	.7	.2	1
5	.5	+1	.2	-.3	2
6	.6	0	.8	.3	2
7	.6	+1	.4	-.1	3
8	.7	+1	.1	-.4	4
9	.8	0	.9	.4	4
10	.8	+1	.7	.2	5
11	.8	+1	.5	.0	6
12	.8	+1	.3	-.2	7
13	.9	+1	.2	-.3	8

FIGURE 3

1.3 Generation of Functions

For schematic purposes, it is convenient to replace the finite increments $(\Delta X)_i$, $(\Delta Y)_i$, and $(\Delta Z)_i$ by the differentials dX, dY, and dZ respectively. Accordingly, the representation of an integrator shown in fig. 1 may be replaced by the schematic shown in fig. 4.

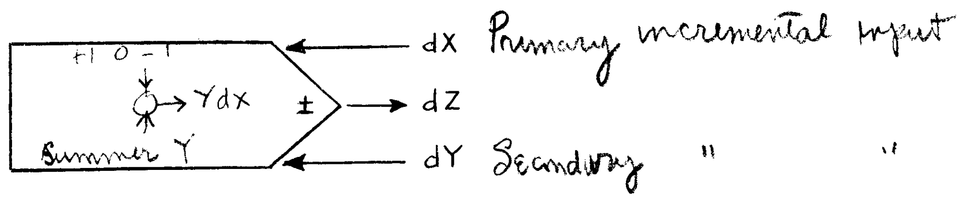


FIGURE 4

The relation between the inputs and the outputs of this integrator is

(Eq. 2a) $dZ = YdX.$

If the output of the integrator is made the secondary input to another integrator, the Y register of the second integrator will accumulate the variable Z.

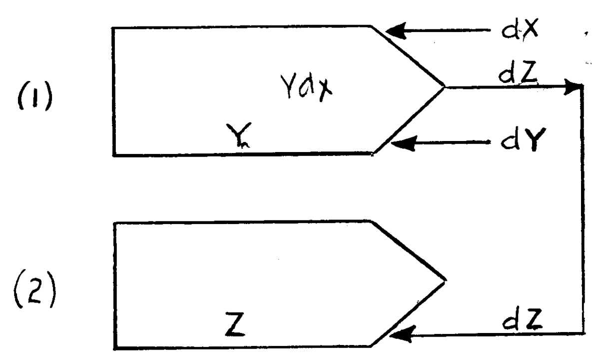


FIGURE 5

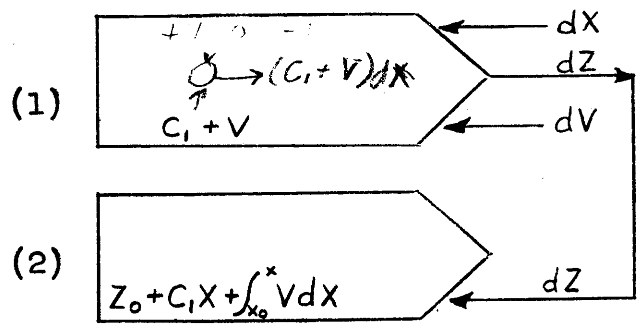
This will take the form

(Eq. 2b) $$Z = \int_{X_0}^{X} Y dX + Z_0.$$

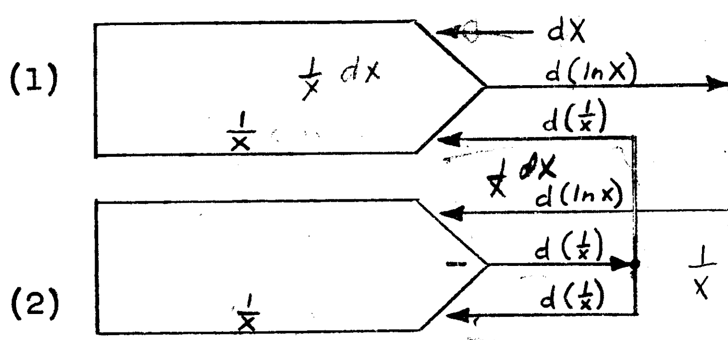
The differential equation associated with a particular interconnection of inputs and outputs may be deduced by means of equations 2a and 2b.

To illustrate the manner in which functions may be generated on a digital differential analyzer, certain integrator hookups will now be presented.

Case I



Case II



$\frac{d}{dx} \ln x = \frac{1}{X}$

$d(\ln x) = \frac{1}{X} dx$

$\frac{d(\frac{1}{X})}{dX} = -\frac{1}{X^2}$

$d(\frac{1}{X}) = -\frac{1}{X^2} dx$

$\frac{1}{X} d(\ln x) = \frac{1}{X} \cdot \frac{1}{X} dx$

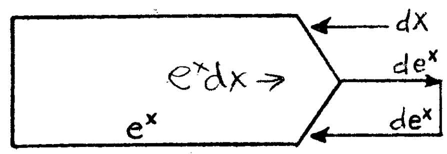
$= \frac{1}{X^2} dx$

$= -d(\frac{1}{X})$

$-\frac{1}{X^2} dx = d(\frac{1}{X})$

With dX as the primary input and $\frac{1}{X}$ as the integrand, the first integrator generates lnX. With d(lnX) as the primary input, $\frac{1}{X}$ as the integrand and an output sign reversal, the second integrator generates $d(\frac{1}{X})$ as the secondary input to both integrators.

Case III

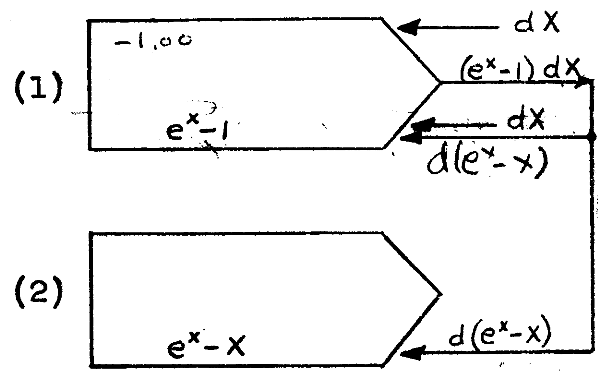


$$\frac{d}{dx} e^x = e^x$$
$$de^x = e^x dx$$

If the output from an integrator is used as its own secondary input, an exponential function, e^X, is generated in the Y register of the same integrator.

Case IV

?

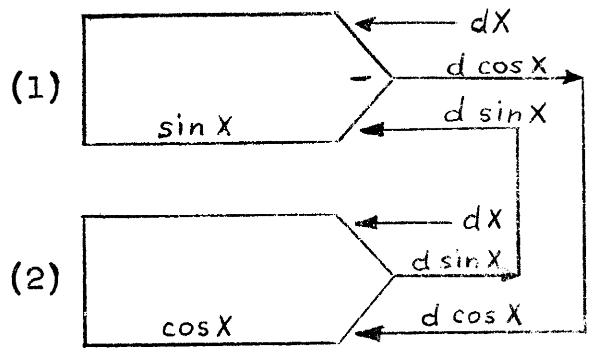


$$d(e^x - x) = de^x - dx = e^x dx - dx = (e^x - 1) dX$$

$$(e^x - 1) dx + dx$$

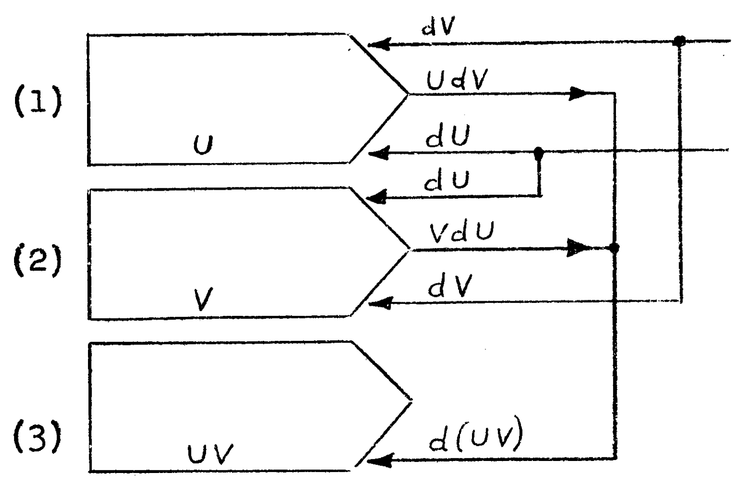
$$dx(e^x - 1 + 1) = e^x dx$$

Case V



Two integrators are being used to generate the sine and cosine of an arbitrary function, X. The minus sign in the first integrator indicates that the sign of the output of that integrator is to be reversed. If the sign reversal were omitted, the hyperbolic functions, sinh X and cosh X, would be generated by means of the same hook-up.

Case VI



The product of two functions may be generated by means of

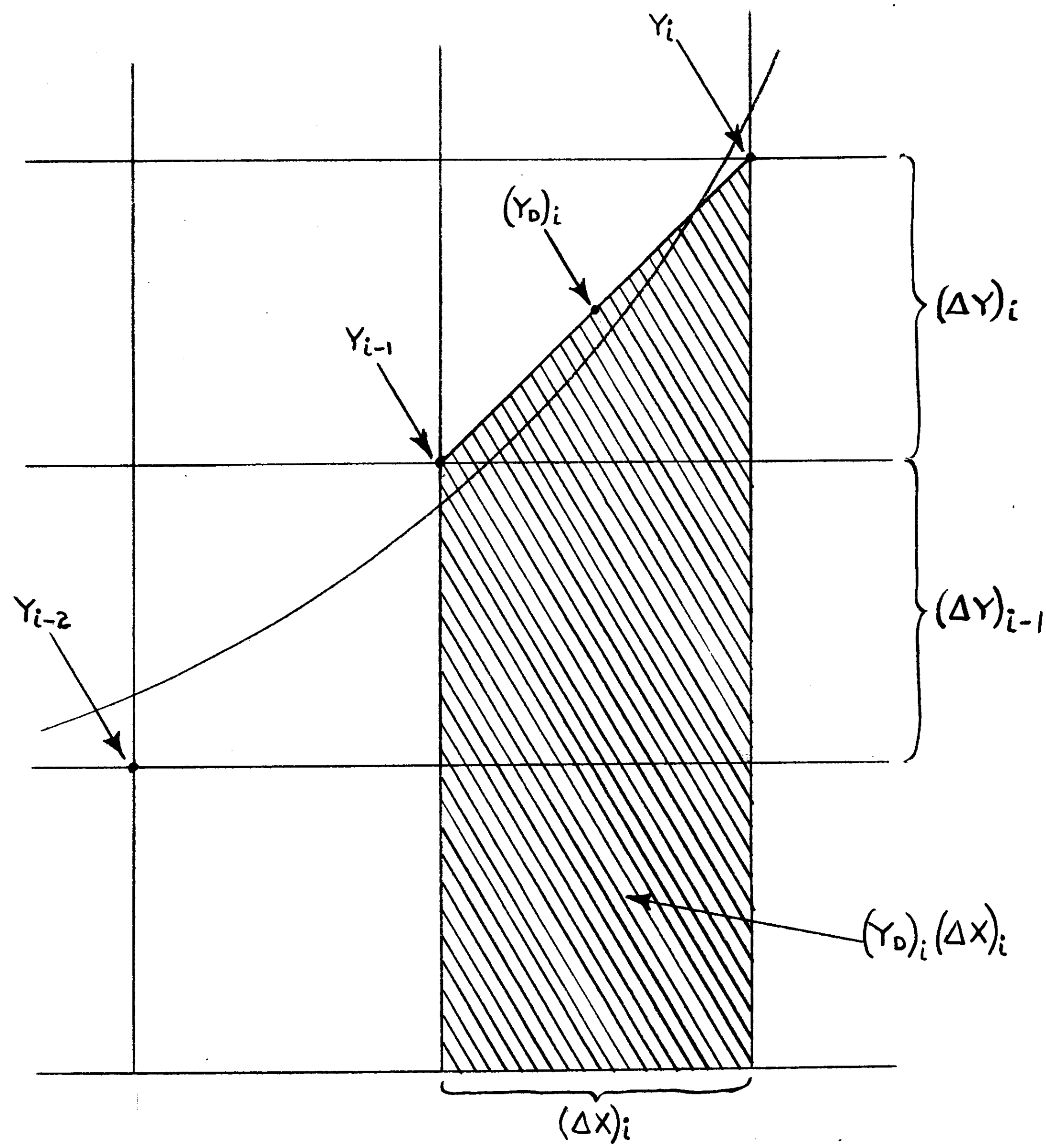
(Eq. 3) $$UV = \int_{U_o}^{U} VdU + \int_{V_o}^{V} UdV + U_o V_o .$$

1.4 Trapezoidal Integration

Up to this point a simple fixed program for performing numerical integrations has been considered, namely, that in which the area under a curve is approximated by summing the areas of a series of rectangles. However, in addition to this method of integration, the Bendix Digital Differential Analyzer employs a fixed program analogous to what is commonly known as trapezoidal integration, in which the area under a curve is approximated by summing the areas of a series of trapezoids (see fig. 6). Here the fundamental area of integration is $(Y_D)_i$ $(\Delta X)_i$, where $(Y_D)_i$ is an estimate of the average value of Y in the interval $(\Delta X)_i$.

Trapezoidal integration effects an important reduction in the error of the approximation in rectangular integration. In fact, the ratio of the error terms in the two methods is of the order of ΔX and, if ΔX is relatively coarse, the reduction in the error is considerable.

1.4.1 Interpolative Mode: In general, some of the successive primary incremental inputs that enter an integrator



Eq. 4 $\quad (Y_D)_i = Y_{i-1} + \frac{1}{2} h_2 (\Delta Y)_i \cong Y_{i-2} + \frac{3}{2} h_2 (\Delta Y)_{i-1}$

FIGURE 6

throughout a number of integration cycles will be zero. Consequently, if an integrator is to perform trapezoidal integration, the program of the integrator must take this into account. One way in which this is accomplished is as follows: A third register connected with the integrator, called a Y_D register, accumulates the quantity $(Y_D)_i = Y_n + \frac{1}{2}h_2 \sum_{n-1}^{i} (\Delta Y)_s$, where Y_n is the value assumed by Y_i when the last non-zero primary incremental input was received and $\sum_{n-1}^{i} (\Delta Y)_s$ is the sum of all the secondary incremental inputs entering the integrator after Y_n is set into the Y_D register. When a non-zero primary incremental input occurs, the quantity in the Y_D register, $(Y_D)_i$, is either added to, or subtracted from, the contents of the R register, in accordance with the sign of the primary input. Thus, the Y value integrated is the average value of Y in the region between the non-zero primary incremental inputs rather than the Y value at either end point. When an integrator is programmed to perform trapezoidal integration in this manner it is said to operate in the Interpolative Mode. An integrator that operates in the Interpolative Mode is indicated by an I written on the left-hand side of the integrator Schematic (see fig. 9).

1.4.2 Extrapolative Mode: The Interpolative Mode only gives greater accuracy when the secondary incremental inputs to an integrator are integrator outputs that have occurred in the same integration cycle.

However, the solution of most problems requires the use of integrator hook-ups that involve integrators whose secondary incremental inputs are integrator outputs that have occurred in the preceding integration cycle. An example of this is the generation of e^X, where the secondary incremental input to an integrator in one integrator cycle is the output from the same integrator in the preceding integrator cycle (see 1.3, Case III). Here, in each cycle, the integrator must be programmed to use as its secondary incremental input an output generated in the preceding cycle. In other words, the secondary incremental input to the integrator is $(\Delta Y)_{i-1}$ during the i-th cycle. Therefore, when trapezoidal integration is to be performed and the primary incremental input is $(\Delta X)_i$, the Y value to be integrated must be predicted by an extrapolation of the $(\Delta Y)_{i-1}$ input. This is accomplished by linear extrapolation (see fig. 6). An integrator programmed to perform such an extrapolation is called an Extrapolative Integrator, and is said to operate in the Extrapolative Mode. Extrapolative integrators are identified by an E written on the left-hand side of the integrator schematic (see fig. 9a). When an integrator operates in this mode it functions in exactly the same way as an integrator operating in the Interpolative Mode, except that the quantity accumulated in the Y_D register during the i-th cycle is

$$(Y_D)_i = Y_{n-1} + \frac{3}{2} h_2 \sum_{n}^{i-1} (\Delta Y)_s$$

instead of $$Y_n + \frac{1}{2}h_2 \sum_{n-1}^{i} (\Delta Y)_s .$$

1.43 Multiplicative Mode: If rectangular integration is used when generating the product of two functions, the hook-up shown in 1.3, Case VI produces

(Eq. 5)
$$\begin{aligned}\Delta(UV) &= \left[V_{i-1} + (\Delta V)_i\right] (\Delta U)_i \\ &\quad + \left[U_{i-1} + (\Delta U)_i\right] (\Delta V)_i \\ &= V_{i-1} (\Delta U)_i + U_{i-1}(\Delta V)_i \\ &\quad + 2(\Delta U)_i (\Delta V)_i ,\end{aligned}$$

and an overlap occurs in which the same area is counted twice (see fig. 7). This error is eliminated by programming the Multiplicative Mode for the two integrators with an 'M' written by the left-side of the integrator schematics (see fig. 9b).

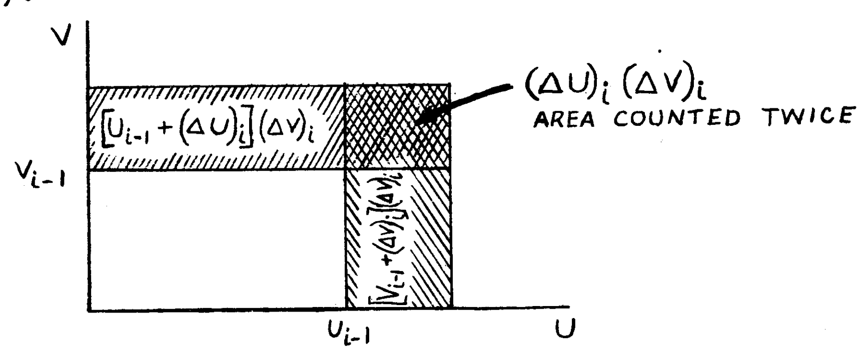


FIGURE 7

In finite increments, the multiplicative mode gives

$$\Delta(UV) = (U + \tfrac{1}{2}\Delta U)\Delta V + (V + \tfrac{1}{2}\Delta V)\Delta U. \qquad \text{(Eq. 6)}$$
$$= U\Delta V + V\Delta U + \Delta U \Delta V$$

The Multiplicative Mode requires the successive secondary incremental inputs entering the integrator from cycle to cycle to be accumulated in the Y Register as before; except that, during a cycle in which the primary incremental input differs from zero, say during the i-th cycle, the current value of the quantity in the Y Register is to be increased by $\frac{1}{2}(\Delta Y)_i$ before being added to, or subtracted from, the contents of the R Register. It should be emphasized that the contents of the Y Register at the end of the i-th cycle is always $Y_i = Y_{i-1} + (\Delta Y)_i$, but that the quantity added to, or subtracted from, the R Register is $Y_{i-1} + \frac{1}{2}(\Delta Y)_i$ when $(\Delta X)_i \neq 0$.

1.5 Special Functions of Integrators

In addition to its use in integration, an integrator may be employed to perform certain other operations. Although these operations do not involve integration, the term "integrator" is retained for the sake of uniformity.

1.5.1 Servos: A servo is an integrator that is programmed to continually make a correction that tends to reduce an error.

what correction

As a mathematical construct, it receives no primary incremental inputs and makes no use of an R Register. If there is no sign reversal in the servo, it simply receives secondary incremental inputs, accumulates them in the Y Register and, depending upon whether the current value thus accumulated is positive, negative, or zero, yields a positive, a negative, or a zero incremental output. The variable of which these outputs are increments is determined by the particular integrator hook-up in which the servo is involved. An integrator which is programmed to operate as a servo is identified by an S written at the left of the integrator schematic.

A servo is used to generate u as a function of v where u is defined implicitly as a function of v by $F(u,v) = 0$. Fig. 8 shows the general operation of a servo in which $\frac{\partial F(u,v)}{\partial u}$ must be non-zero and the output sign of the servo must be the opposite of the sign of $\frac{\partial F(u,v)}{\partial u}$. Under these conditions, when incremental changes in v cause $F(u,v) \neq 0$, the servo will generate incremental changes in u until $F(u,v) = 0$. Figure 9a shows a hook-up for the case in which $F(u,v) = e^u - v$ and fig. 9b shows the case when

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

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

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

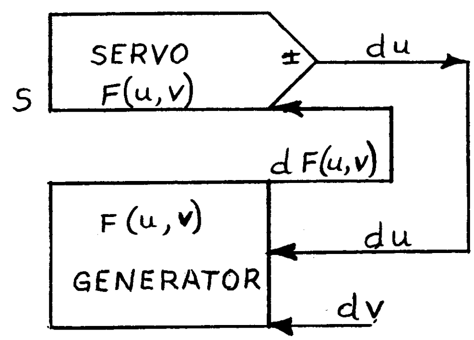


FIGURE 8

1.5.2 Adders: An adder is an integrator which operates like a servo except that one of its secondary inputs is the negative of its output. An integrator which operates as an adder is identified by an A written at the left of the integrator schematic.

Fig. 10a 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 $x + w - u = 0$.

Fig. 10b shows the connections of a servo to obtain the negative of the sum of the variables, x and w.

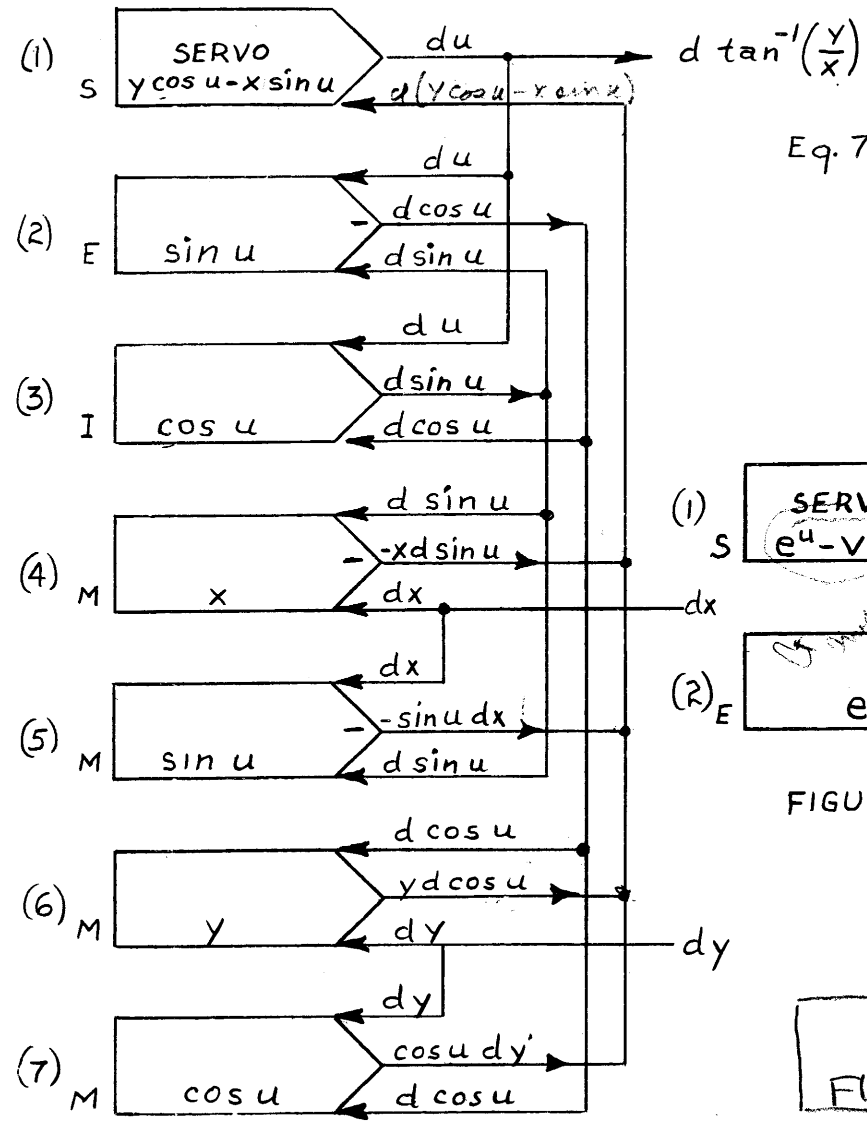


FIGURE 9b

Eq. 7 $\quad u = \tan^{-1} \frac{y}{x}$

$e^u = v \qquad u = \ln v$

$du = d(\ln v)$

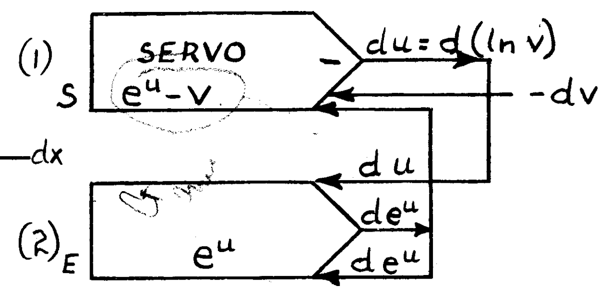
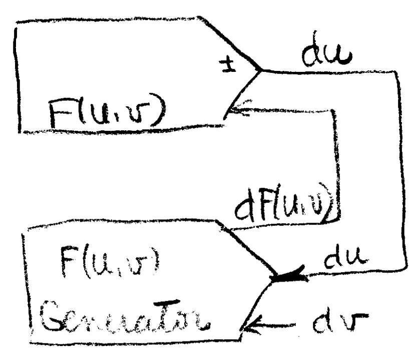


FIGURE 9a



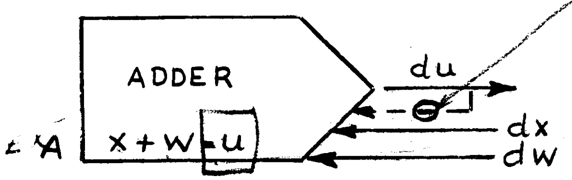


FIGURE 10a

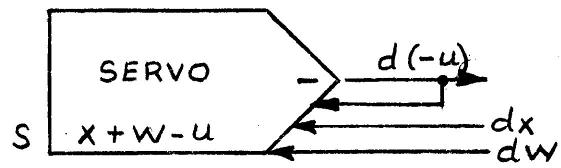


FIGURE 10b

1.5.3 <u>Decision Integrators</u>: Decision integrators are used to generate a wide variety of discontinuous and non-linear functions, such as square waves, saw-tooth waves, etc. Like a servo, a decision integrator does not employ an R Register; unlike a servo, it often receives primary incremental inputs as well as secondary incremental inputs. An integrator programmed as a decision integrator is identified by an S written at the left of the integrator schematic, and by the word "Decision" written across the schematic. The successive secondary incremental inputs to a decision integrator are accumulated in its Y register from cycle to cycle. However, in any cycle, say the i-th cycle, the incremental output, $(\Delta Z)_i$, is determined by the primary incremental input, $(\Delta X)_i$, and the quantity currently accumulated in the Y Register, Y_i, in accordance with the following scheme:

(Eqns 8)

a) If $+2 > Y_i \geqslant +1, \; (\Delta Z)_i = 0$

b) If $+1 > Y_i > 0, \; (\Delta Z)_i = (\Delta X)_i$

c) If $Y_i = 0, \; (\Delta Z)_i = 0$

— 0.3 — 3

(Eqns 8 - cont'd) d) If $0 > Y_i > -1$, $(\Delta Z)_i = -(\Delta X)_i$

e) If $-1 \geqslant Y_i \geqslant -2$, $(\Delta Z)_i = 0$

Figures 11, 12, and 13 show three integrator hook-ups that involve decision integrators. In fig. 11 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, using properties b) and d) of its scheme of operation, emits the absolute value of u as its incremental output. In fig. 12 two decision integrators are being used to generate the saw-tooth functions, w and v, by means of the same properties.

Figure 13 illustrates how a decision integrator may be used to simulate a clipped sine wave by means of properties a) and b) of its scheme of operation.

1.6 Output Multipliers

The relative frequency of non-zero incremental outputs emitted by an integrator is called its output rate. The maximum output rate occurs when an integrator emits a non-zero output during every cycle of operation.

Frequently, the quantity accumulated in a Y Register has a known upper bound. If this bound has a value of .5 or less,

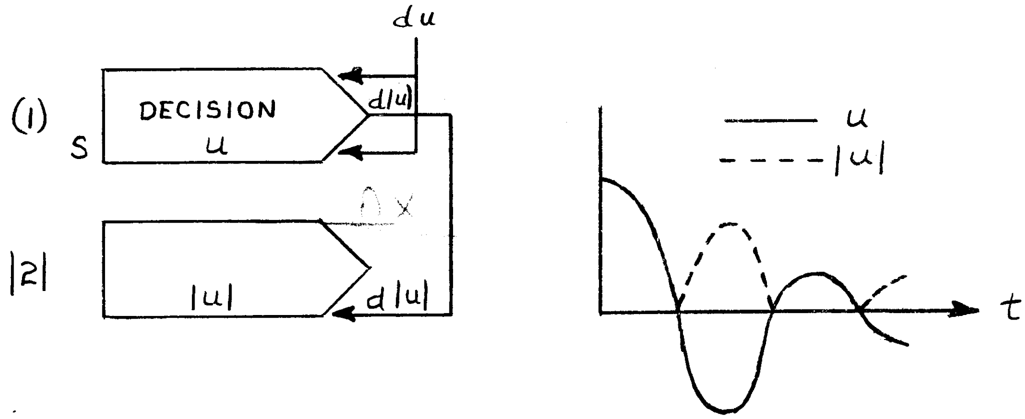
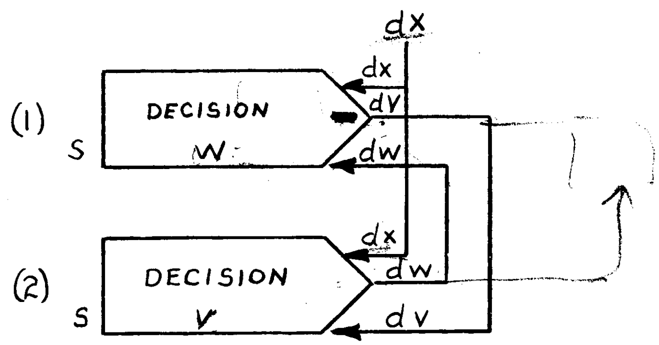


FIGURE 11



Eqns. 9

$$\frac{dv}{dx} = -\operatorname{Sgn}[w]$$

$$\frac{dw}{dx} = \operatorname{Sgn}[v]$$

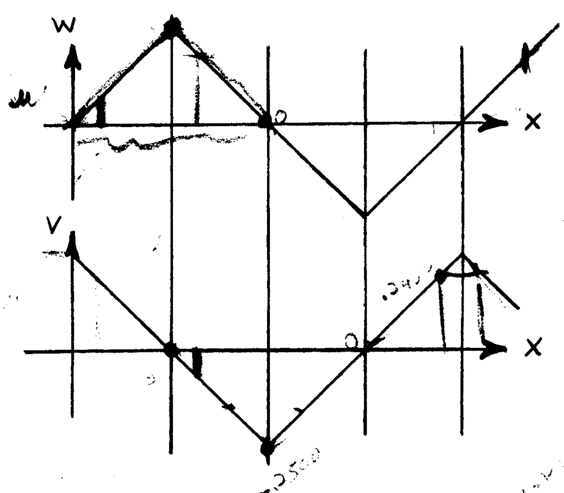


FIGURE 12

Eqns. 10

$$u(\theta) = b \sin\theta \quad \text{when } b\sin\theta \leq a$$

$$u(\theta) = a \quad \text{when } b\sin\theta \geq a$$

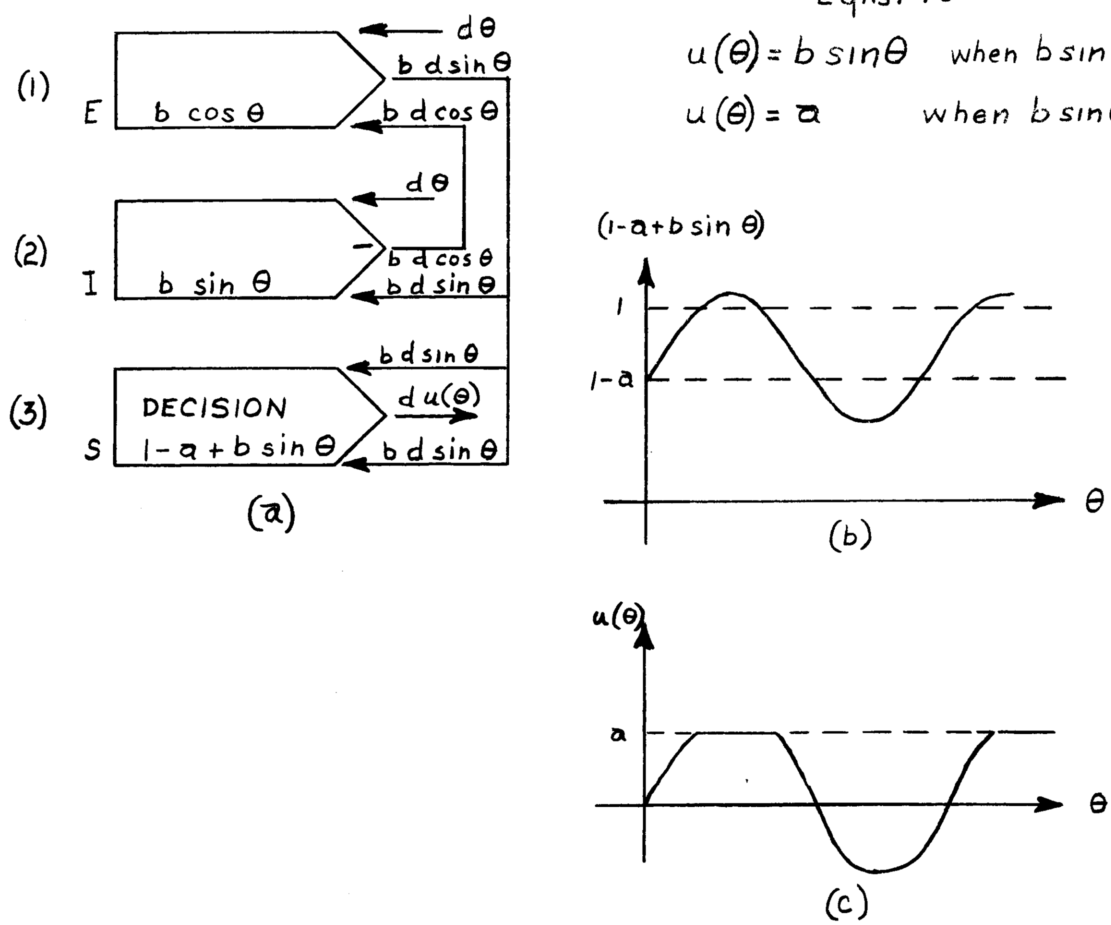


FIGURE 13

it is apparent that the output rate of the associated integrator will not be over 1/2 the maximum rate; i.e., the R Register will not overflow more than half the time. Similarly, if the upper bound has a value of .2 or less, the output rate will only be 1/5 the maximum rate, or only one out of every 5 cycles of operation will produce a non-zero incremental output. Consequently, there would be a considerable increase in the efficiency with which an integrator performs its function if these output rates could be increased. Within the limitations imposed by the scaling of the problem, the output rate is increased by restricting the capacity of the R Register, thereby increasing the frequency of its overflow. In the Bendix Digital Differential Analyzer, the output rate of an integrator may be multiplied by 2 or 5, thus respectively requiring the quantity in the Y Register to be less than .5 or .2, and restricting R_1 as follows:

for multiplication by 2; $0 \leq R_1 < .5$, $R_o = .25$

for multiplication by 5; $0 \leq R_1 < .2$, $R_o = .1$

The use of an output multiplier of 2 or 5 is indicated in the integrator block by a 2 or 5 for M following the output sign (see fig. 14).

1.7 Variation and Restoration of Initial Conditions

An outstanding feature of the Bendix Digital Differential

Analyzer is its ability to vary automatically the initial conditions of a problem in accordance with results obtained during computation. When a problem is coded into the computer, initial values must be entered in the Y and Y_D registers of all participating integrators before the commencement of the first integration cycle. These values are determined from that particular solution of the equations being solved which corresponds to a desired starting point for computation.

In each integrator, the associated initial value is not only entered in both the Y and Y_D registers, but is also stored in an additional register, called an Initial Condition register, Y_I. The initial condition register may be used either to accumulate the incremental outputs from any given integrator, thereby permitting an initial value to be computed for use in obtaining a succeeding solution, or, if this has not been done, to recommence computation by reentering the original initial values in the Y and Y_D registers.

The last integrator, number 59,* is called the Reset Control Integrator, and it differs from the others in its ability to

*When the computer is operated as a thirty instead of a sixty integrator machine, the last integrator is numbered 29 and becomes the reset control integrator.

perform an additional operation. This integrator emits a signal whenever the value accumulated in its Y register attains a magnitude of one. This signal causes the Y and Y_D registers in a preselected group of participating integrators to be set to the current values stored in their respective initial condition registers.* Integrators that have been selected to be coded for this resetting are identified in diagrams by placing an R to the left of the integrator block (see fig. 16). The variable or constant written at the top of the integrator block indicates the quantity to be stored in the initial condition register, while the symbol at the upper left-hand corner indicates an incremental input, ΔY_I (in fig. 14, indicated by dY_I) to the initial condition register.

The manner in which the computer may be used to solve problems through the variation of initial conditions will now be illustrated by means of two examples:

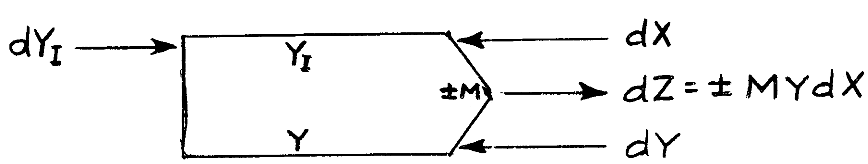


FIGURE 14

*Whenever the quantity in the initial condition register is entered in the Y and Y_D registers of an integrator, the content of the R register of that integrator is set to R_o.

Example I

(Eq. 11) $$d^2y/dx^2 = Ax + e^y + C$$

$$\dot{y}_o = -B \qquad \dot{y}_f = 0$$

$$y_o = ? \qquad y_f = 0$$

$$x_o = 0 \qquad x_f = ?$$

This problem involves a split boundary condition. It is solved on the computer by finding the value, y_o, and then generating that particular curve $y(x)$, that satisfies both sets of conditions (see fig. 15).

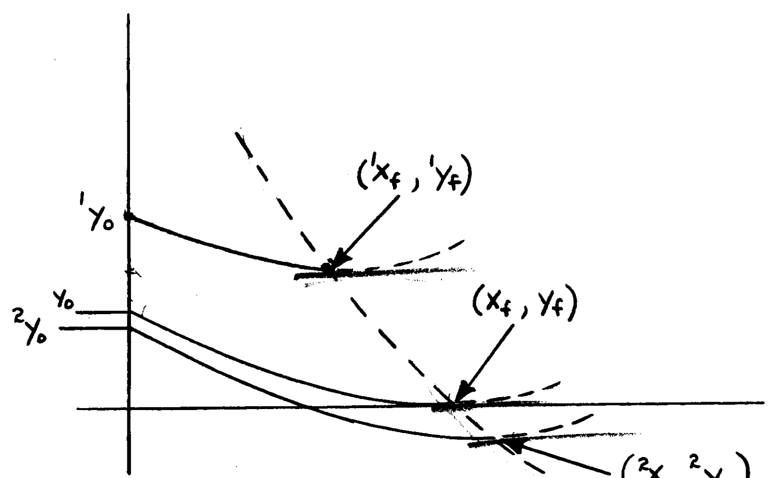


FIGURE 15

Initially, a trial estimate, ^1y_o, is made and the corresponding solution is generated to the point $({}^1x_f,\ {}^1y_f, {}^1\dot{y}_f = 0)$. Successive initial conditions are then computed that make

(Eq. 12) $${}^{n+1}y_o = {}^ny_o - m({}^ny_f),$$

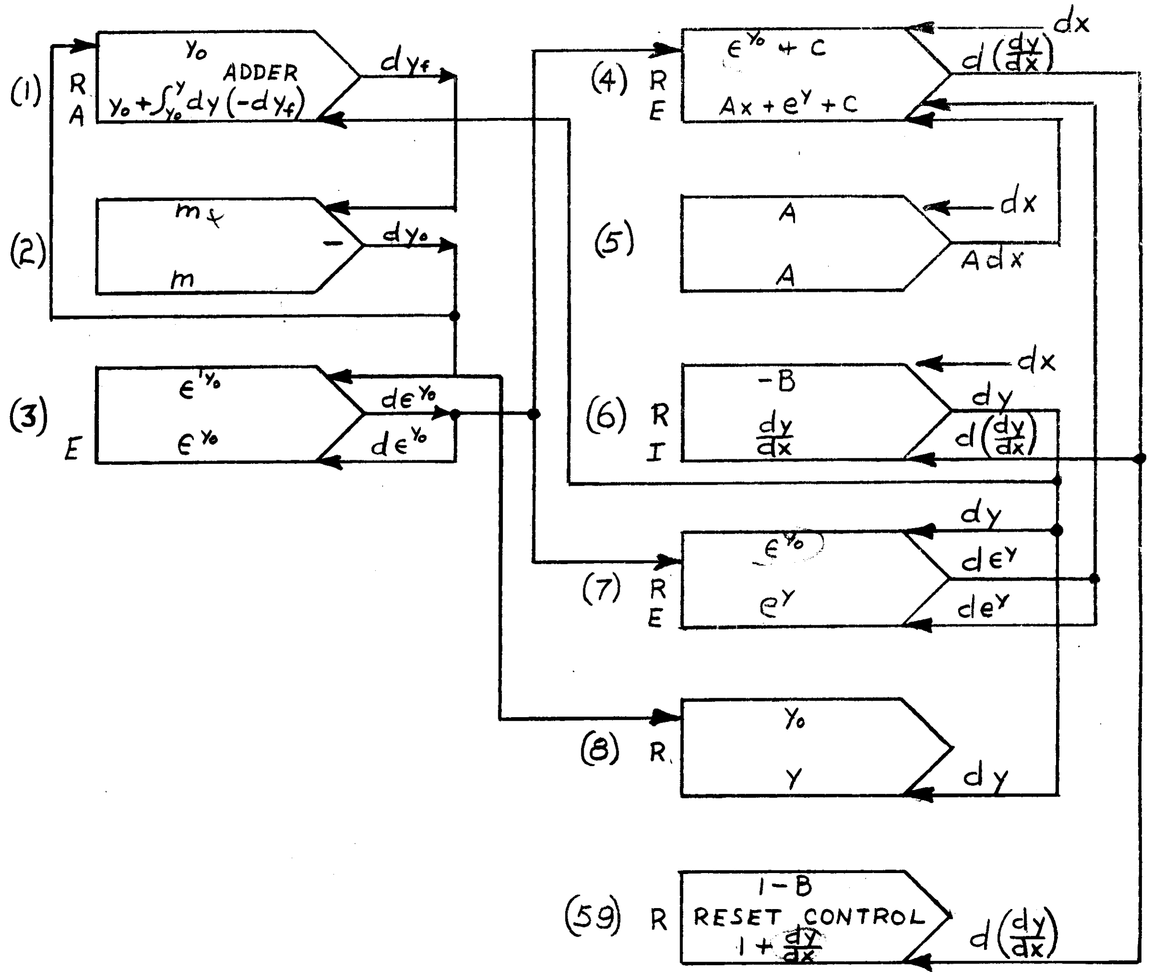


FIGURE 16

$\frac{dy}{dx} = 0$

$1 - B$

$1 - 2 = -1$

where m is a damping coefficient that only needs to be changed if ${}^{i}y_f$ diverges from zero.

The computation may be made to automatically converge to the solution by means of an integrator hook-up such as that shown in fig. 16.

Integrators (1), (2), and (3) are used to generate a variation in the initial condition that will decrease the next $|{}^{i}y_f|$. Integrators (4), (5), (6), (7), and (8) are used to generate y(x) for each estimate of y_o. Integrator (59) signals the computer to introduce the corresponding initial condition each time dy/dx becomes zero.

With the setting of each initial condition, the value of the current estimate of y_o is entered in the Y register of the adder integrator (1), where it is augmented by the incremental changes that occur during the trial solution. Consequently, as the i-th trial solution is generated, the resulting total incremental output from integrator (1) equals ${}^{i}y_f$.

Example II:

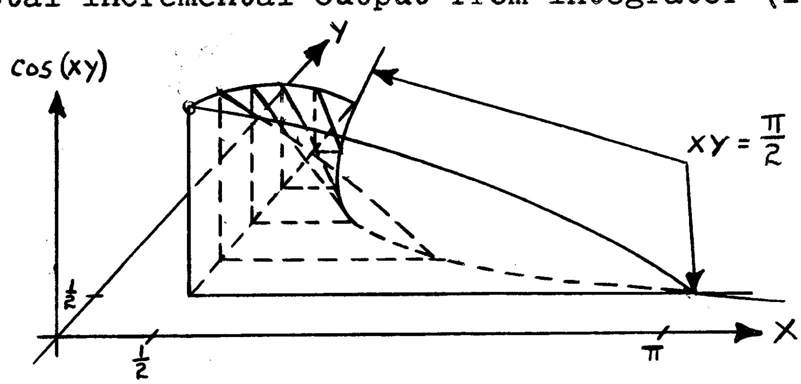


FIGURE 17

(Eq. 13) $$V = \int_{1/2}^{\pi} \int_{1/2}^{\pi/2y} \cos (xy) \, dydx$$

The evaluation of this simple double integral in eq. 13 may be accomplished by summing the areas of a number of slabs. Say the n-th slab has a thickness Δy and an area on one side equal to $\int_{1/2}^{\pi/2y_n} \cos (y_n x)dx$. If there are m slabs,

(Eq. 14) $$y_1 = 1/2, \; y_2 = 1/2 + \Delta y, \ldots, y_n = 1/2 + (n-1)\Delta y, \ldots \ldots, y_m = 1/2 + (m-1)\Delta y = \pi.$$

Now, the initial value of cos (xy) for each slab is cos (y/2). This function of y is generated by integrators (1), (2) and (3) in fig. 18. The incremental change of y from y_n to y_{n-1} is generated by the adder, integrator (1). With each signal from the reset control integrator, (59), $1/2 \Delta y$ is entered in the Y register of integrator (1), thereby causing this integrator to emit positive increments at the maximum rate until the quantity 1/2y has been increased by $1/2 \Delta y$. When this happens, the value in the Y register of integrator (1) is zero. The corresponding incremental changes in 1/2y, cos y/2, and sin y/2 are directed as incremental inputs to the initial condition registers of integrators (4), (5), (6), (7), and (59), where they are accumulated to form new initial conditions. When a signal is received from the reset control integrator (59),

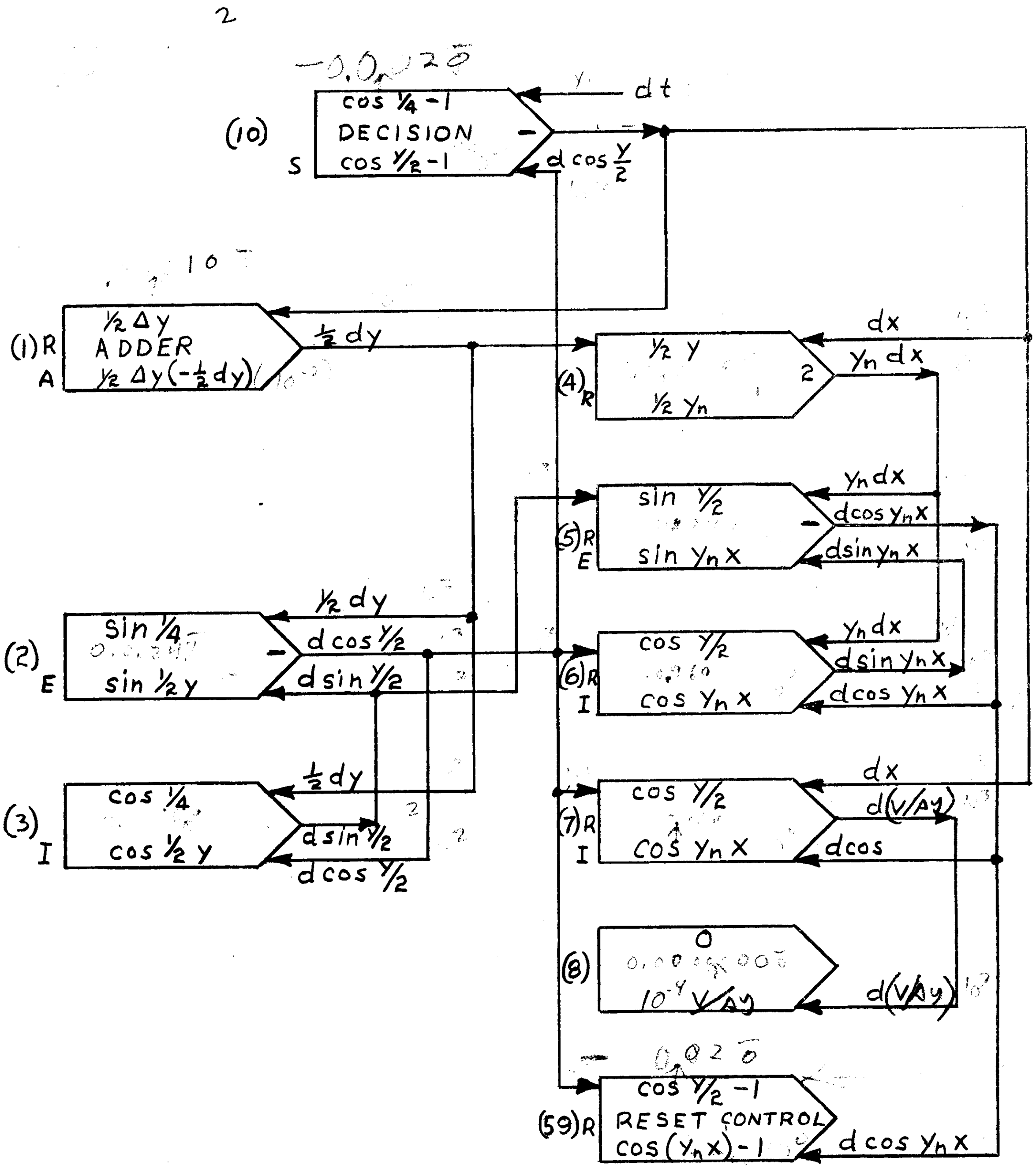

(10)
S
cos ¼ -1
DECISION
cos Y/2 -1
dt
d cos Y/2
(1) R
A
½ Δy
ADDER
½ Δy (-½ dy)
½ dy
(2)
E
sin ¼
sin ½ y
½ dy
d cos Y/2
d sin Y/2
(3)
I
cos ¼
cos ½ y
½ dy
d sin Y/2
d cos Y/2
(4) R
½ y
½ yn
2
dx
yn dx
(5) R
E
sin Y/2
sin yn x
yn dx
d cos yn x
d sin yn x
(6) R
I
cos Y/2
cos yn x
yn dx
d sin yn x
d cos yn x
(7) R
I
cos Y/2
cos yn x
dx
d(V/Δy)
d cos
(8)
0
10^-4 V/Δy
d(V/Δy)
(59) R
cos Y/2 -1
RESET CONTROL
cos (yn x) -1
d cos yn x

FIGURE 18

these new conditions are entered in the appropriate Y and Y_D register and the area of the corresponding slab is computed in a straightforward manner. The sum of all these areas is formed in integrator (8). Integrator (59), automatically emits a signal whenever cos(xy) -1 equals minus one. This corresponds to the limit of integration, $x = \pi/2y_n$. Integrator (1) emits the incremental outputs used to advance the integration until the quantity, cos(y/2) -1, contained in its Y register equals minus one. Consequently, computation is effectively halted when y equals π.

1.8 Input and Output

Several forms of input and output are provided with the computer. These involve punched tape equipment, an electric typewriter and incremental plotters.

1.8.1 Typewriter Output: Any integrator except the last one, integrator 59, (or 29), may be programmed for typeout control with a 6 written in the output multiplier position. A non-zero output of a typeout control integrator signals the computer to type out the integrands of all integrators (except the last one) programmed for typeout with a T written by the left side of the integrator schematics. Generally the integrand of a typeout control integrator will be a constant $(\frac{1}{n})$ so that the integrator will cause typeouts to occur at intervals of n primary incremental inputs. This means that each primary incremental input of plus one will cause $\frac{1}{n}$ to be added into the R register resulting, after n such inputs, in one output and a typeout. The number of functions that may be typed out will have an upper limit between 14 and 28 depending on the precision of the numbers being typed.

When typewriter readout is used during computation, a tape may be punched automatically with the same information that is typed. If only one integrator is programmed for typeout and the information is punched on tape as well as typed, the tape obtained may be used as a function input.

1.8.2 Input and Output of Non-incremental Functions: The computer is provided with means for filling the computer with problems from punched tape. This tape filling mechanism may also be used for input functions in one of two ways.

The first method is used to enter more than one function of the same independent variable. The tape contains the destination integrator numbers and the corresponding integrand values to be entered at each of the equally spaced values of a monotonic variable generated by the computer. The values to be entered for each point are filled from the tape into the initial condition registers of the destination integrators which are also programmed for automatic resetting. The incremental variations in the input control variable are used as the secondary input to the last integrator and when its integrand attains a magnitude of one, a resetting operation occurs.* After each entry, the initial condition registers of the destination integrators are filled from the tape again with the values corresponding to the next succeeding point.

The second method of integrand input allows only one function to be entered but the function controlling the input can be any variable generated by the computer. The tape used in this case contains only the values for each point of entry and the destination is automatically integrator 2. The last integrator controls the entry of the values in essentially the same way as the first method. However, when the control variable reverses directions, special means are provided for backing the tape to fill the proper entry into the initial condition register of integrator 2.

The program for using these integrand inputs will depend on the nature of the values entered. If the entries are the zero, first or second differences of the desired function, they

*The R registers are not altered by this operation.

could be used as shown in figs. 19a, 19b, or 19c.

1.8.3 Incremental Input and Output: The incremental outputs from twelve integrators are routed to the input connections of six possible incremental type graph plotters.

Plotter No.	x	vs	y
1	0		1
2	18		19
3	20		21
4	22		23
5	24		25
6	57(or 27)		58(or 28)

For example, the first plotter plots the output of integrator 0 against the output of integrator 1 as x versus y. When these integrators are used to generate plotter outputs, their integrands usually contain scaling constants to obtain the desired size of graphs. These scaling constants will be discussed in section 1.9.2.

These incremental outputs may also be used for other purposes such as providing incremental inputs to a second computer. For this purpose, each computer has provision for accepting up to eight incremental inputs from an external source. Except for plotter outputs, the use of these incremental inputs and outputs can only be discussed in terms of the use

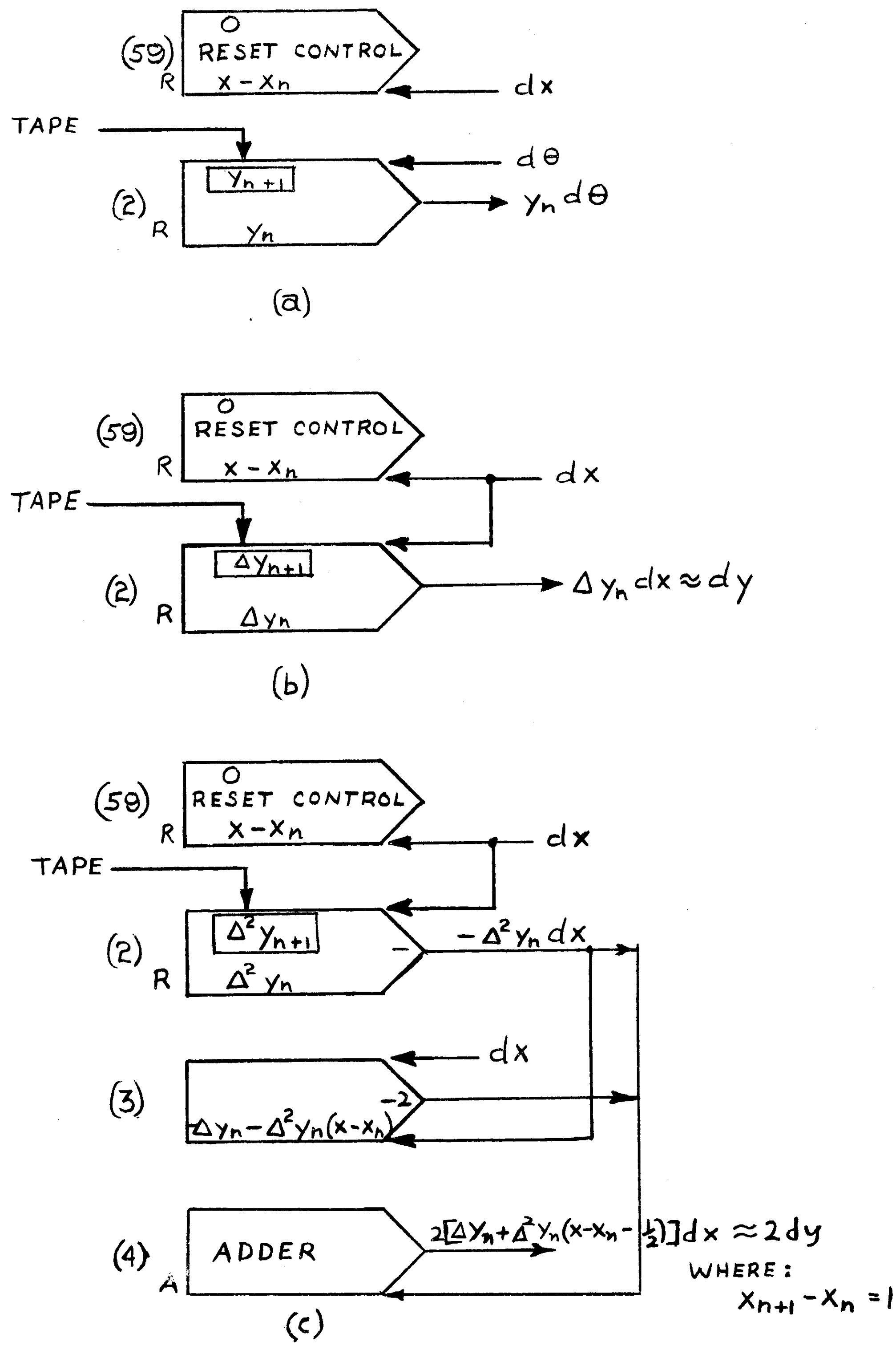

(59)
O
RESET CONTROL
R
X - Xn
dx
TAPE
Yn+1
dθ
(2)
R
Yn
Yn dθ
(a)
(59)
O
RESET CONTROL
R
X - Xn
dx
TAPE
ΔYn+1
(2)
R
ΔYn
ΔYn dx ≈ dy
(b)
(59)
O
RESET CONTROL
R
X-Xn
dx
TAPE
Δ2Yn+1
(2)
R
Δ2Yn
−
−Δ2Yn dx
dx
(3)
−2
ΔYn−Δ2Yn(X-Xn)
(4)
A
ADDER
2[ΔYn+Δ2Yn(X-Xn-½)]dx ≈ 2dy
WHERE:
Xn+1 - Xn = 1
(c)

FIGURE 19

that is made of them in a given computer installation.

A special problem arises when the output of an adder is to be used as an input to a plotter. For example, a simple adder (section 1.5.2), when used to add the outputs of two integrators, can have two, near simultaneous, non-zero outputs when the integrator outputs are non-zero simultaneously or in close succession. If the two close non-zero outputs from the adder would be of the same sign only, a small enough scaling constant, say C_1 in integrator 1, could be chosen to reduce the adder output rate for use as any input to plotter number 1. However, when the two close non-zero outputs are of opposite sign, integrator 1 can also have two, close, non-zero outputs of opposite sign. No choice of C_1 can be made to completely eliminate this latter form of output to plotter number 1. A special filter-adder for adding two integrator outputs is shown in fig. 20.

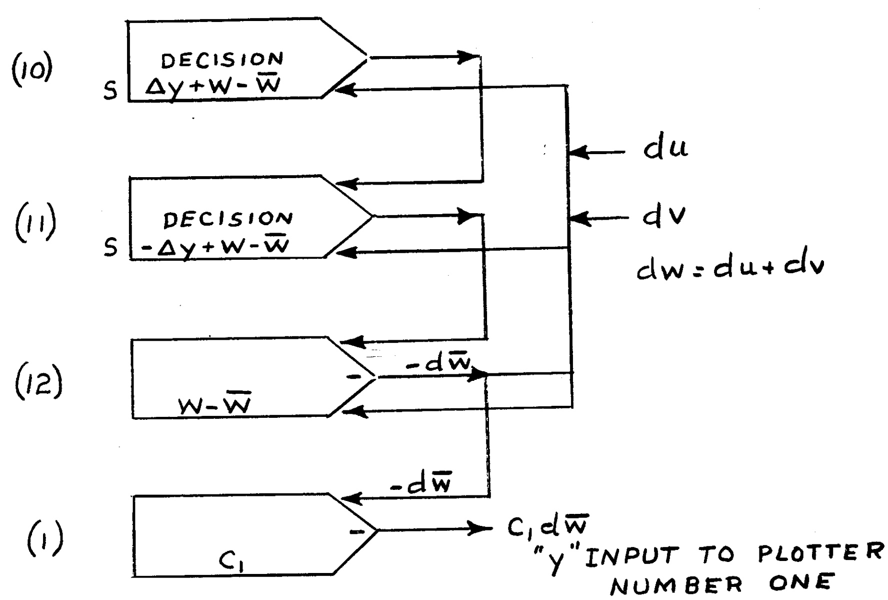


FIGURE 20

Integrators 10, 11 and 12 generate a function, $-\overline{W}$, with no close non-zero increments. $\overline{W}$ is "stable" when $|W - \overline{W}| \leq$ one increment in W and, when $\overline{W}$ lags W by more than one increment, $\overline{W}$ is incrementally corrected at rate proportional to the lag. Integrators 10, 11 and 12 can be visualized as operating in series so that if any one of these integrands is zero, $\overline{W}$ will receive no incremental change. Integrator 10 blocks $\overline{W}$ increments when $W - \overline{W}$ equals one negative increment, and integrator 11 blocks $\overline{W}$ increments when $W - \overline{W}$ equals one positive increment. When $|W - \overline{W}| \neq$ one increment, integrator 12 varies $\overline{W}$ at a rate proportional to $(W-\overline{W})$. Assuming a reasonable continuity in the integrator outputs being summed, the "stable region" in the $\overline{W}$ generation eliminates close $\overline{W}$ increments of opposite sign and the proportional operation of integrator 12 eliminates close $\overline{W}$ increments of the same sign.

When three to eight integrator outputs are to be summed for plotting, the "stable region" must be broadened. Fig. 21 shows a filter adder for n integrator outputs with the "stable region" extended to $|W - \overline{W}| \leq (n-1)$ increments in W.

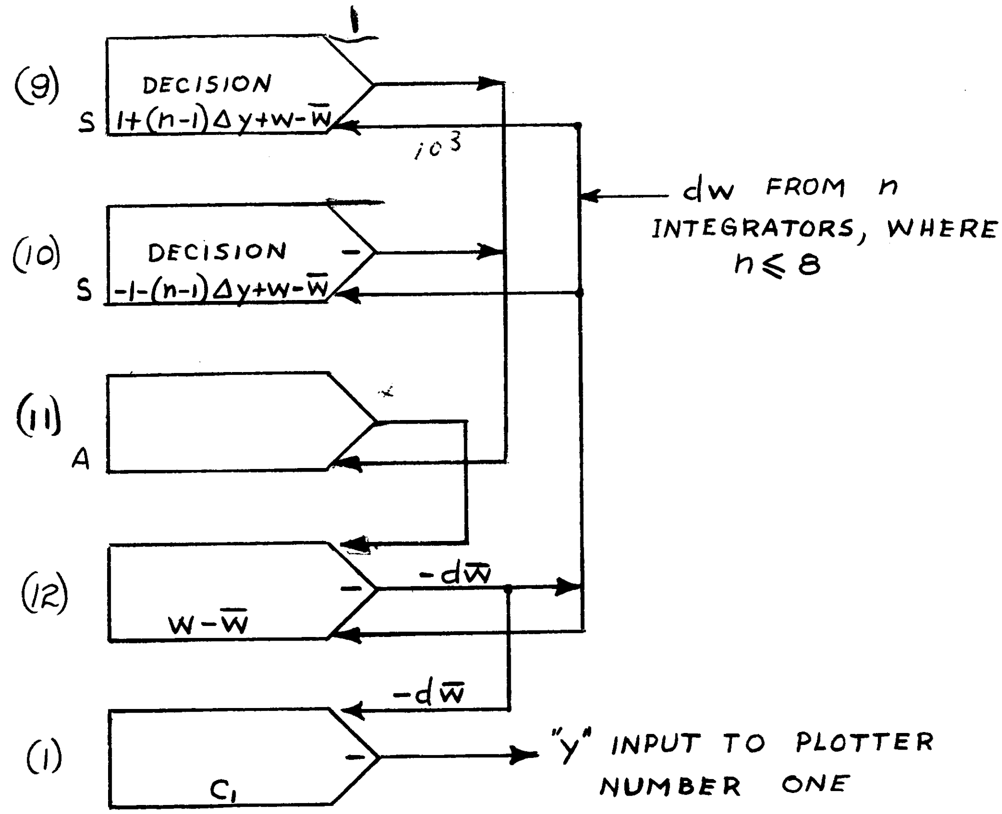


FIGURE 21

Integrator 9 will have a positive output each time $(W-\overline{W}) \leqslant -n$ increments, and integrator 10 will have a positive output each time $(W-\overline{W}) \geqslant n$ increments. Whenever either integrator 9 or 10 has an output, it is transmitted by the adder in integrator 11 to integrator 12 which varies $\overline{W}$ at a rate proportional to $(W-\overline{W})$.

1.8.4 Problem Read-in and Read-out: Problems are usually prepared for entry into the computer by using the typewriter to operate a tape punch. The form of the copy to be typed while punching this filling tape will be described in section 1.9.3. The punched tape is fed through a tape reader to enter the problem into the computer. Problems can also be partly or completely entered into the computer from a manual keyboard.

The complete state of a problem can be read out of the computer onto a punched tape. The information on this tape may then be refilled into the computer at any later time. The facilities for reading out a problem onto punched tape are, in general, used for two purposes. First, it may happen that after filling a problem from a tape, trial runs indicate errors in coding. After these have been righted manually a corrected punched tape copy of the initial conditions of the problem may be desired. Rather than punch a new tape from the typewriter, the computer can automatically copy all the information in the computer onto a tape in a form which, while not that of the automatic filling procedure, can be re-entered into the computer. More important, these facilities may be used to copy all the information held in the computer at some point after the start of computation so as to render unnecessary, in the event that computation must be interrupted, the complete rerunning of a problem.

1.9 Programming

The Programming of problems for solution by the computer involves three steps: Mapping, Scaling and Coding. The variety of problems and methods of solving them is so great that only a general approach to programming will be presented.

The steps in programming an example problem are presented at the end of the following three sections. The equation of this problem is

(Eqn. 15) $$\ddot{x} + k\,(x^2 - 1)\,\dot{x} + x = 0.$$

Equation 15 represents a voltage loop in an oscillator circuit. The resistance term, $k(x^2 - 1)$, is negative when $x^2 < 1$ and positive when $x^2 > 1$. While the resistance is negative, the energy in the loop is increased and while the resistance is positive the energy is decreased. For each value of k, equation 15 has a stable oscillatory solution such that the energy gains and losses cancel. These stable solutions are often represented by plotting $\dot{x}$ vs. x, as shown in fig. 22.

1.9.1 Mapping: Mapping consists in setting up an integrator information flow diagram which corresponds to the problem to be solved. Several such problem maps have already been presented. When the problem is one expressed as

Eq. 15

$$\ddot{x} + k(x^2 - 1)\dot{x} + x = 0$$

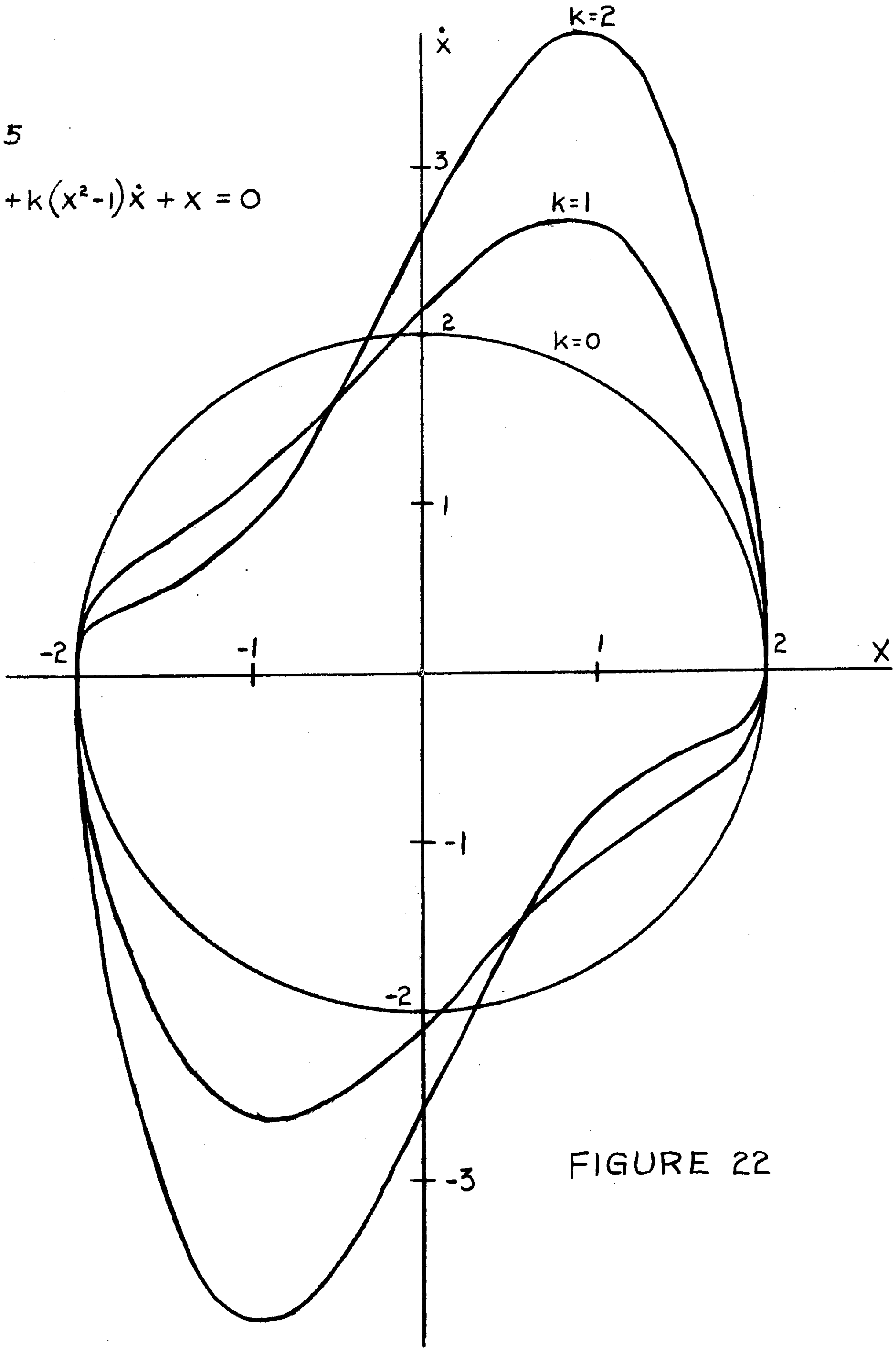


FIGURE 22

a single n-th order differential equation,

$$f(t, \theta, \frac{d\theta}{dt}, \ldots, \frac{d^n\theta}{dt^n}) = 0,$$

it is best to solve the equation for the highest derivative,

$$\frac{d^n\theta}{dt^n} = g(t, \theta, \frac{d\theta}{dt}, \ldots, \frac{d^{n-1}\theta}{dt^{n-1}}).$$

This later equation may then be multiplied through by dt giving the equation,

$$d\left(\frac{d^{n-1}\theta}{dt^{n-1}}\right) = g(t, \theta, \frac{d\theta}{dt}, \ldots, \frac{d^{n-1}\theta}{dt^{n-1}})dt$$

which may then be simply mapped as follows:

1. Let the integrand of the first integrator be $\frac{d^{n-1}\theta}{dt^{n-1}}$ and integrate it against t obtaining $d(\frac{d^{n-2}\theta}{dt^{n-2}})$ as the output.

2. Let the integrand of the second integrator be $\frac{d^{n-2}\theta}{dt^{n-2}}$ and integrate it against t obtaining $d(\frac{d^{n-3}\theta}{dt^{n-3}})$ as the output.

3. Continue this chain as indicated in fig. 23.

4. Use the outputs of these integrators along with dt to generate $d(\frac{d^{n-1}\theta}{dt^{n-1}})$ which is required as the secondary input to the first integrator.

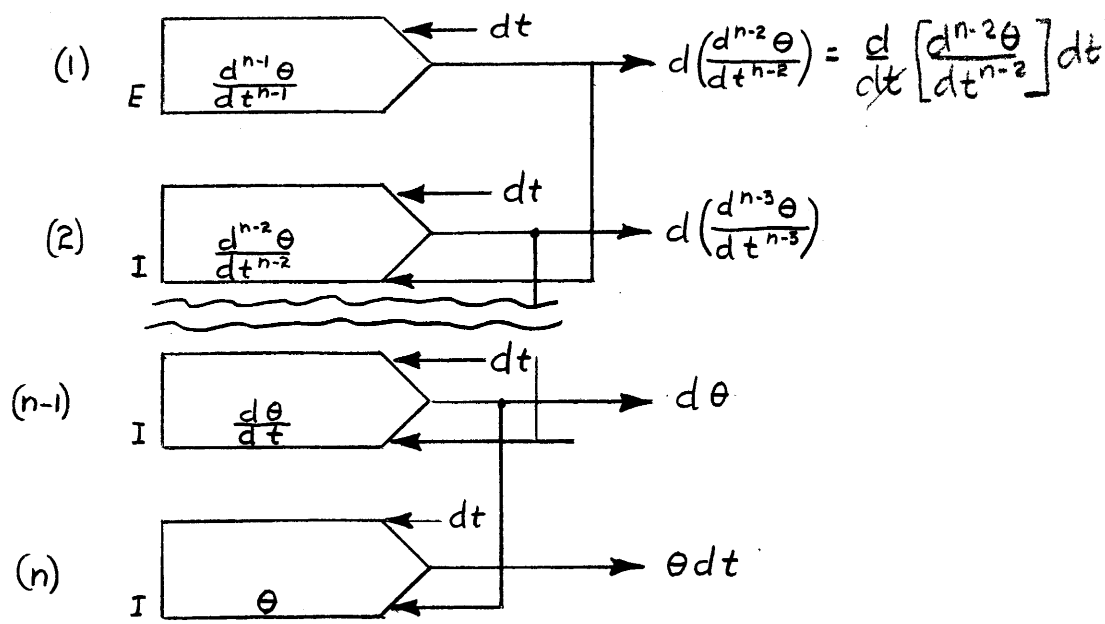


FIGURE 23

A simple illustration of this type of map is shown in fig. 24.

A similar procedure may be used in mapping a problem involving a simultaneous set of differential equations, as shown in fig. 25.

(Eqn. 17)

$$\frac{d^3W}{dt^3} = W \frac{dV}{dt} + V$$

$$\frac{d^3V}{dt^3} = W \frac{dW}{dt} + Vt$$

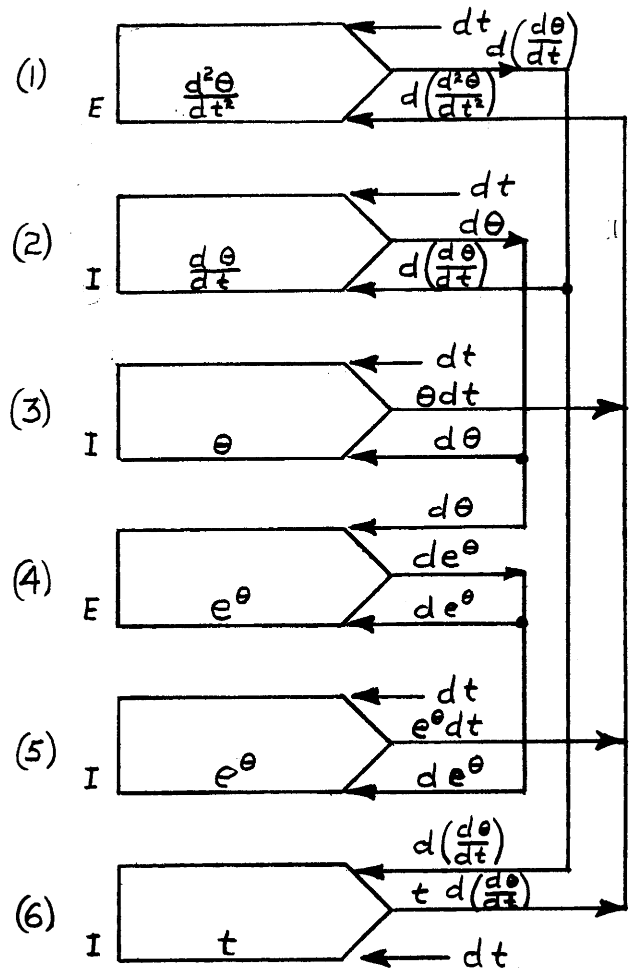


FIGURE 24

Eqns. 16

$$\frac{d^3\theta}{dt^3} = t\frac{d^2\theta}{dt^2} + e^\theta + \theta$$

$$d\left(\frac{d^2\theta}{dt^2}\right) = t\,d\left(\frac{d\theta}{dt}\right) + e^\theta dt + \theta dt$$

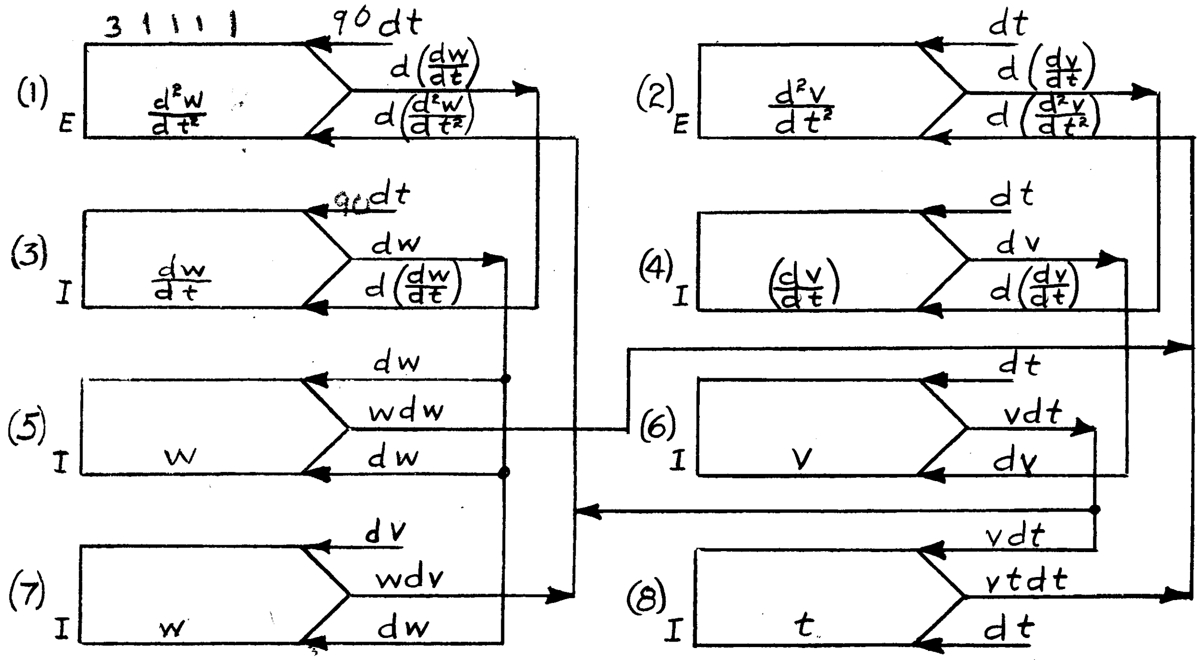


FIGURE 25

It may happen that the described scheme for mapping will give rise to an integrand which becomes infinite at some point in the problem. For the equation, $Y'' = -\frac{1}{x}Y' - Y$, the map shown in fig. 26a cannot be used when x is small or zero. Fig. 26b shows an alternative map that may be used in the region when x is small. Integrator 3 is coded as a servo which generates $\frac{1}{x}\,dY$ as its output.

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 fig. 27.

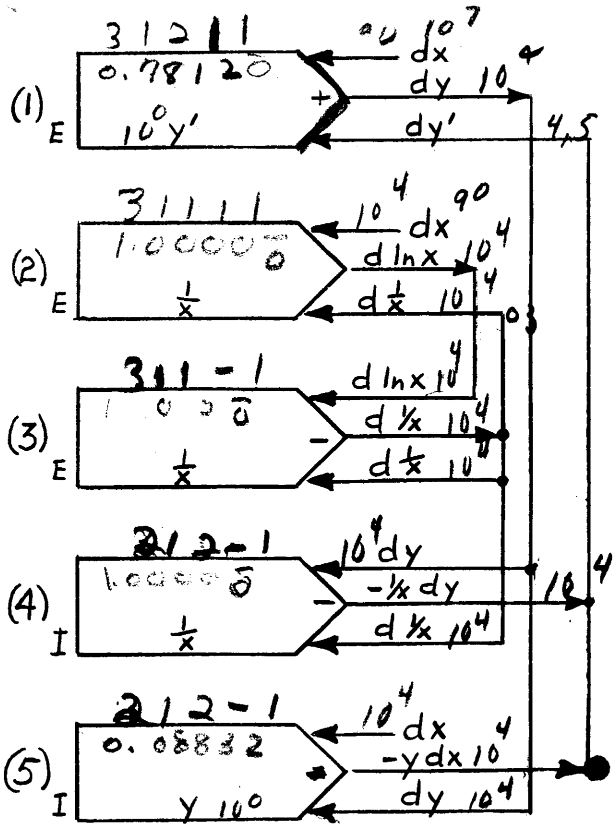


FIGURE 26 a

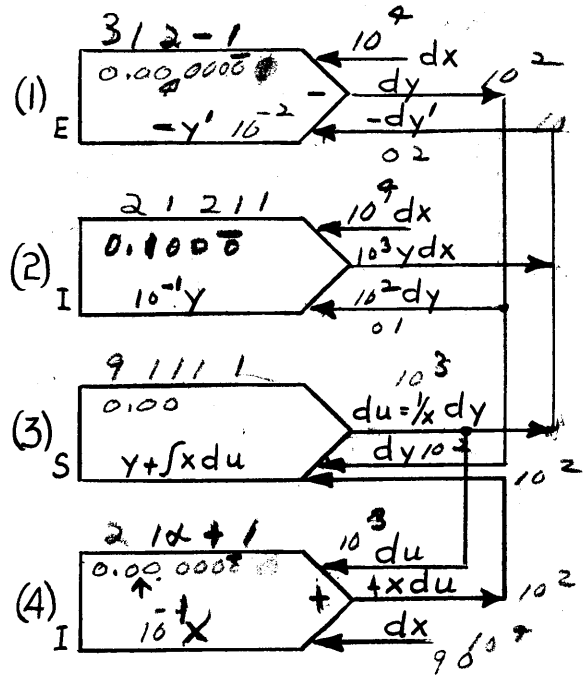


FIGURE 26 b

Eqns. 18

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

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

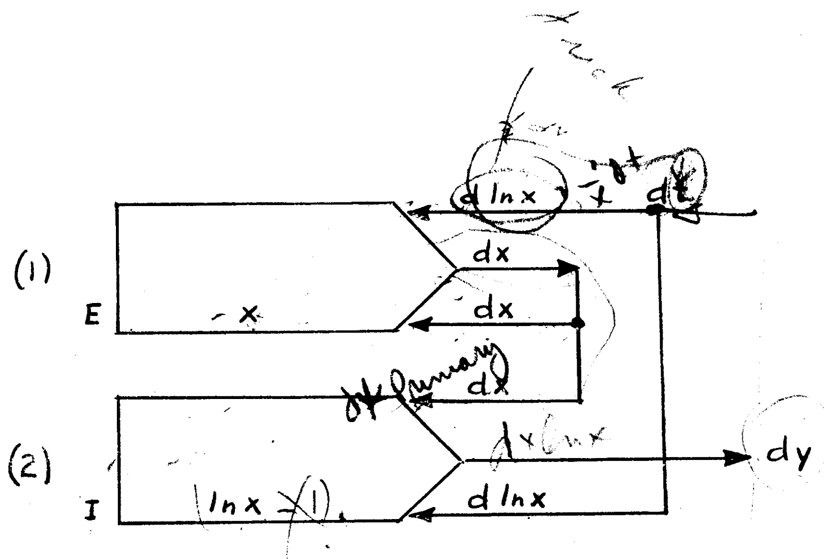


FIGURE 27

The thing to appreciate is that the machine independent variable is not always used as the independent variable of the problem. The machine independent variable may even be used as the problem dependent variable as shown in fig. 28.

Eqns. 19

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

$dx = xy\,dy$

$y^2 = 2 \ln x$

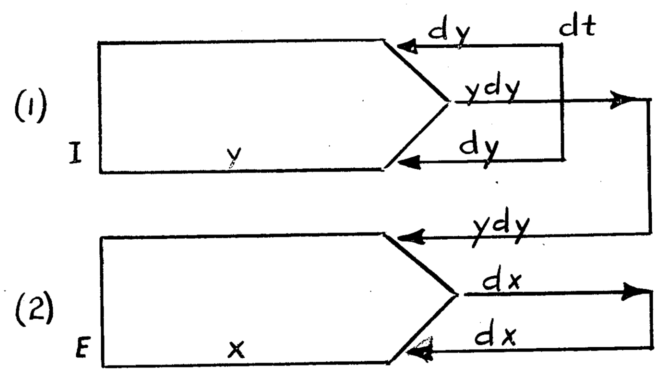


FIGURE 28

The map shown in fig. 28 can be used with the initial condition, $x = 1$, $y = 0$, even though $(\frac{dy}{dx})_o$ is infinite.

The maps shown this far have had integrator numbers assigned as well as the modes of integration (I for interpolative, E for extrapolative, and M multiplicative). The assignment of integrator numbers determines the order in which each integrator will perform its part of the solution for each iteration. Each iteration begins with the operation of integrator number 0 followed by the operation of each integrator in numerical order until the last integrator operates which concludes the iteration. The assignment of the modes of integration is determined by this order of integration and the flow of information indicated by the map. The following rules may be followed in assigning integrator numbers and modes.

1. The primary input to each integrator should be the machine independent variable or the output of a smaller numbered integrator.

2. The secondary input should also be the machine independent variable or the output of a smaller numbered integrator whenever possible. Integrators programmed in this way should be assigned the interpolative mode.

3. When the secondary input to an integrator is its own output or the output of a greater numbered integrator, the extrapolative mode should be used.

4. An integrator with more than one secondary input should have all of its secondary inputs come from the sources

as described in either rule (2) and be programmed for the interpolative mode or rule (3) and be programmed for the extrapolative mode. If rule (2) is used, nine secondary inputs are allowed, but if rule (3) is used, six secondary inputs are allowed.

5. An adder is allowed eight secondary inputs which should be the outputs of smaller numbered integrators. No mode is assigned, but an A is placed by the integrator schematic.

6. A servo is allowed nine secondary inputs but has no primary input. No mode is assigned but an S is placed by the integrator schematic. Very often the use of a servo will require the violation of some of the preceding rules but the amount of violation should be kept to a minimum.

7. A decision integrator is distinguished from a servo only in that it will usually have a primary input and that its integrand will vary over a greater range.

8. When multiplication of two variables is to be programmed (see sec. 1.4.3), the multiplying integrators should both receive their inputs from smaller numbered integrators. The multiplicative mode is indicated by an M placed by the integrator schematic.

The form of mapping presented becomes cumbersome on larger problems. A more compact notation is illustrated below for the example in fig. ~~22.~~ 24

(Eqns. 20)

1) $d\mathring{\theta} = (\overset{\circ\circ}{\theta})\,dt$

2) $d\theta = (\mathring{\theta})\,dt$

3) $\theta dt = (\theta)\,dt$

4) $de^{\theta} = (e^{\theta})d\theta$

5) $e^{\theta}dt = (e^{\theta})dt$

6) $td\mathring{\theta} = (t)\,d\mathring{\theta}$

The extension of this notation to indicate servos, adders, variable initial conditions, etc. can be made easily.

The map of the example is shown in fig. 29. Integrators 4, 5, 6, 7 and 8 are used to do the basic problem. Integrator 9 is used for typeout control for the integrands of integrators 3, 4 and 5. Integrators 0 and 1 are used to obtain the curve of x vs. t on plotter number 1. Integrators 18 and 19 are used to obtain the curve of $\mathring{x}$ vs. x on plotter number 2. Integrators 10, 11 and 12 form a filter-adder to obtain a plotter input from the outputs of integrators 7 and 8.

1.9.2 Scaling: The numerical values associated with a problem must be scaled so as to produce numbers which the

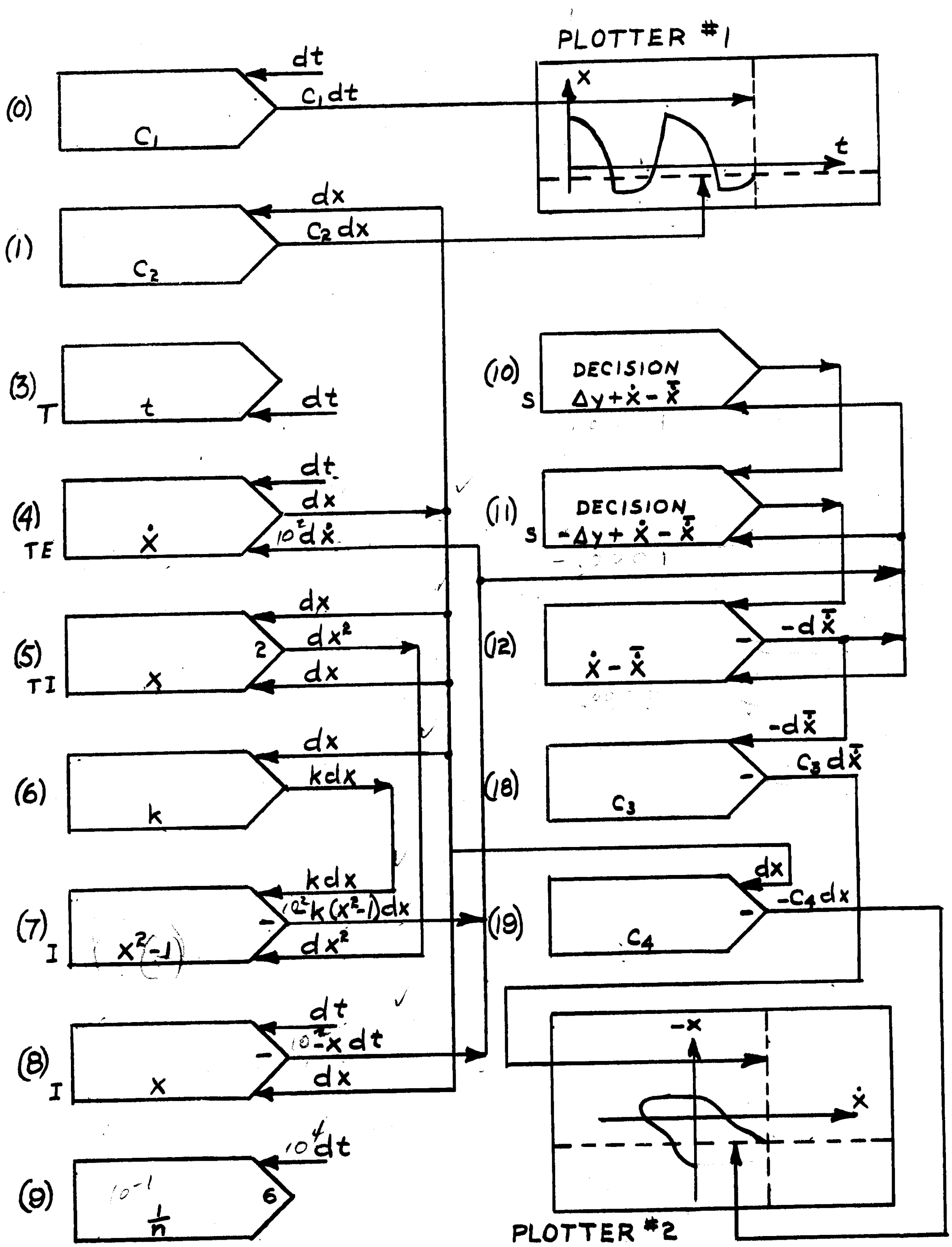


FIGURE 29

Eqns. 15 $\ddot{x}+k(x^2-1)\dot{x}+x=0$ $\qquad d\dot{x}=-k(x^2-1)dx-x\,dt$

computer will process. The scaling relationships involving the computer values (Y, dY, dX and dZ) and the corresponding problem values (Y_p, dY_p, dX_p and dZ_p) will be developed in this section.

All of the scale factors can be defined according to the following equations:

(Eqns. 21) $Y = K_Y Y_p$ $dX = K_X dX_p$

$dY = K_Y dY_p$ $dZ = K_Z dZ_p$

(where all K's are positive)

As previously defined dY, dX and dZ, when they occur as the outputs of single integrators, can each have the values of +1, 0, or -1; and Y can have a maximum of $\frac{1}{M}$, where M is the integrator output multiplier of 1, 2 or 5. The incremental scale factors are, therefore, such that if $K_X = 10^3$, a dX of +1 corresponds to a dX_p of 10^{-3}. The integrand scale factor is such that, if Y_p reaches a maximum of 17 during a problem, K_Y can be 10^{-2} and M be 5.

Formally stated, K_Y and M must be chosen such that

(Eqn. 22) $1 \geq M K_Y |Y_p|$ max.

By substituting the right-hand terms of equations 21 into the equation, $dZ = M YdX$, the following equation is obtained.

(Eqn. 23) $$K_z\, dZ_p = M\,(K_Y\, Y_p)(K_x\, dX_p)$$

and, in order for dZ_p to equal $Y_p\, dX_p$, equation 23 reduces to

(Eqn. 24) $$K_z = M\, K_Y\, K_x$$

The integrand register of each integrator has seven decimal places to the right of the decimal point and a units digit which also indicates the sign of Y as shown in fig. 30.

	. . . .	
positive Y values	1 . 0 0 0 0 0 0 1	= $1 + 10^{-7}$
	1 . 0 0 0 0 0 0 0	= 1
	0 . 9 9 9 9 9 9 9	= $1 - 10^{-7}$
	. . . .	
	0 . 0 0 0 0 0 0 1	= $+10^{-7}$
	0 . 0 0 0 0 0 0 0	= 0
negative Y values	9 . 9 9 9 9 9 9 9	= -10^{-7}
	. . . .	
	9 . 0 0 0 0 0 0 1	= $-1 + 10^{-7}$
	9 . 0 0 0 0 0 0 0	= -1
	8 . 9 9 9 9 9 9 9	= $-1 - 10^{-7}$
	. . . .	

FIGURE 30

Negative values of Y appear in the Y register as complements. The complement of a given value for Y is obtained by subtracting the value from 10.

For some special integrator functions Y may vary over the greater range indicated in fig. 31.

positive Y values	4 . 9 9 9 9 9 9 9	=	$5 - 10^{-7}$
	4 . 0 0 0 0 0 0 0	=	4
	3 . 0 0 0 0 0 0 0	=	3
	2 . 0 0 0 0 0 0 0	=	2
	1 . 0 0 0 0 0 0 0	=	1
	0 . 0 0 0 0 0 0 0	=	0
negative Y values	9 . 0 0 0 0 0 0 0	=	-1
	8 . 0 0 0 0 0 0 0	=	-2
	7 . 0 0 0 0 0 0 0	=	-3
	6 . 0 0 0 0 0 0 0	=	-4
	5 . 0 0 0 0 0 0 0	=	-5

FIGURE 31

When the units digit is less than 5, the value of Y is positive and, when the units digit is 5 or greater, the value of Y is negative.

In addition to having seven digits to the right of the decimal point, the Y register has an extra digit for round-off purposes. The round-off digit is never varied by a dY input to the Y register. However, this round-off digit will be varied by ±5 in the Y_D register when the interpolative mode of integration is used and a dY input is, say, +1 (dY_D = +.5). This round-off digit is not regarded as significant since it is never varied in the Y register during computation and is never typed out but it does allow constants to be expressed to eight significant digits.

In general, N significant decimal places of the Y register will be used where $1 \leq N \leq 7$. With $K_Y Y_p$ held in an N digit register it would take $K_Y Y_p 10^N$ non-zero dY_p inputs of opposite sign to that of Y_p to reduce Y_p to zero. The size of these dY_p inputs would be K_y^{-1}. Therefore, in order for the dY_p inputs to alter Y_p properly, the following equation must be true.

(Eqn. 25) $$(K_Y Y_p 10^N) K_y^{-1} = Y_p$$

or $$10^N = K_y K_Y^{-1}$$

or $$N = \log K_y - \log K_Y$$

and, of course, N must be an integer.

The necessary scaling relationships for an integrator have now been established and are restated below

(Eqns. 26) 1) $1 \geq M K_Y |Y_p| \max \geq -1$

(Eqns. 26 - cont'd) 2) $K_z = M\, K_Y K_x$

3) $1 \leq N = \log K_y - \log K_Y \leq 7$

(where $M = 1$, 2, or 5)

Regarding equation 26-1, it can be seen how $M\, K_Y\, Y_p$ is the fraction of the non-zero dX inputs which can be transmitted as non-zero outputs from an integrator and that this fraction is limited to 1 by the computer's method of operation. It can further be seen how $(M\, K_Y)$, which is limited by the inverse of $|Y_p|$ max, multiplies the fineness of dX_p, namely K_x, to give the fineness of dZ_p, namely K_z.

It will be convenient in many cases to express the scale factors in the following form.

(Eqns. 27)

1) $K_Y = k_Y\, 10^{S_Y}$

2) $K_y = k_y\, 10^{S_y}$

3) $K_x = k_x\, 10^{S_x}$

4) $K_z = k_z\, 10^{S_z}$

(where $1 \leq k < 10$ and the S's are restricted to integers).

Equation 26-3 can be replaced by

(Eqns. 28)

1) $k_y = k_Y$

2) $S_y - S_Y = N$

If, for example, $K_Y = 2\text{x}10^2$, $K_y = 2\text{x}10^6$ and $Y_p = .004832$, the Y register will be as shown in fig. 32.

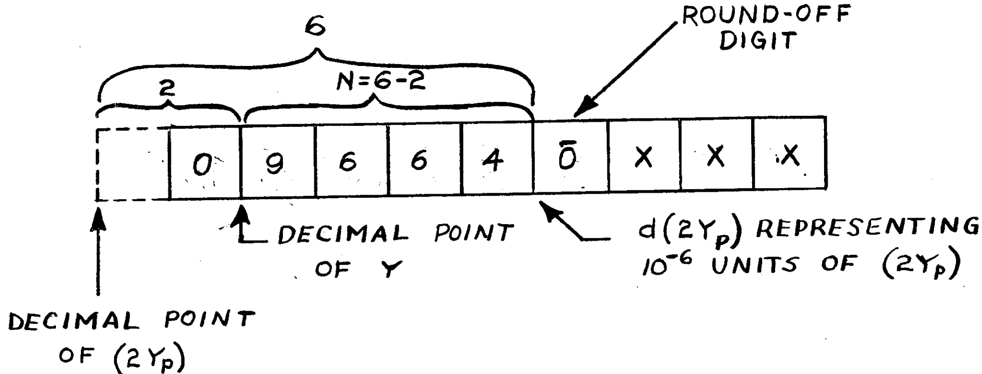


FIGURE 32

$K_Y = 10^{-3} \times .26531$

If, in another case, $K_Y = 10^{-3}$, $K_y = 10^2$, and $Y_p = -265.31$, the Y register will be as shown in fig. 33.

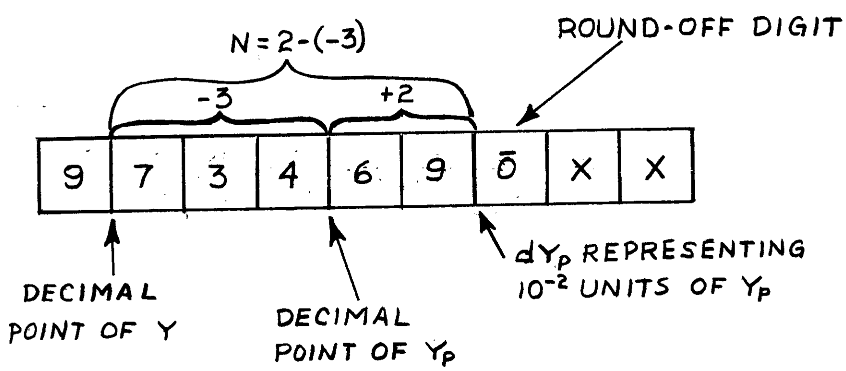


FIGURE 33

Since the decimal point of $k_Y Y_p$ is moved S_Y decimal places to the right to obtain Y and the $d(k_y Y_p)$ increments are added into the Y register S_Y decimal places to the right of the decimal point of $k_Y Y_p$, it can be seen directly that k_y must equal k_Y and $S_y - S_Y = N$.

Y, dY, dX and dZ can be replaced in the integrator schematic as shown in fig. 34,

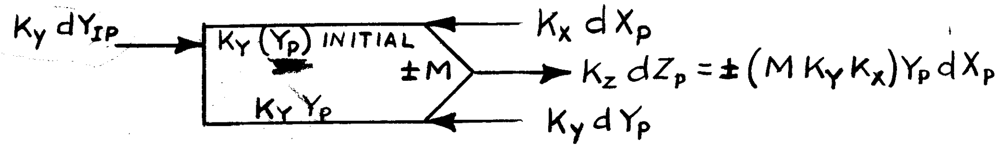


FIGURE 34

It will be convenient to indicate K_Y times the initial value of Y_p in the upper portion of the integrator schematic. The scale factor of an input to the initial condition register must be equal to K_Y as shown in fig. 34.

Integrators 4, 5, 6, 7 and 8 of the problem to be programmed are shown in map form in fig. 35.

$|\dot{X}|$ attains a maximum of about 4 and $|X|$ reaches about 2.02 for k equal to 2. Using these figures, the relationships for determining the scale factors can be written as follows:

$$1 \geqslant M_4\, K_4\, 4 \qquad K_1 = M_4\, K_4\, K_t$$
$$1 \geqslant M_5\, K_5\, 2.02 \qquad 2K_2 = M_5\, K_5\, K_1$$
$$1 \geqslant K_6\, 2 \qquad K_3 = 1\;\; K_6\, K_1$$
$$1 \geqslant M_7\, K_7\, 3.0804 \qquad K_9 = M_7\, K_7\, K_3$$
$$1 \geqslant M_8\, K_8\, 2.02 \qquad K_9 = M_8\, K_8\, K_t$$

$$k_9 = k_4 \qquad 1 \leq S_9 - S_4 = N_4 \leq 7$$
$$k_1 = k_5 \qquad 1 \leq S_1 - S_5 = N_5 \leq 7$$
$$k_2 = k_7 \qquad 1 \leq S_2 - S_7 = N_7 \leq 7$$
$$k_1 = k_8 \qquad 1 \leq S_1 - S_8 = N_8 \leq 7$$

(M_4, M_5, M_7 and M_8 can each individually be 1, 2, or 5)

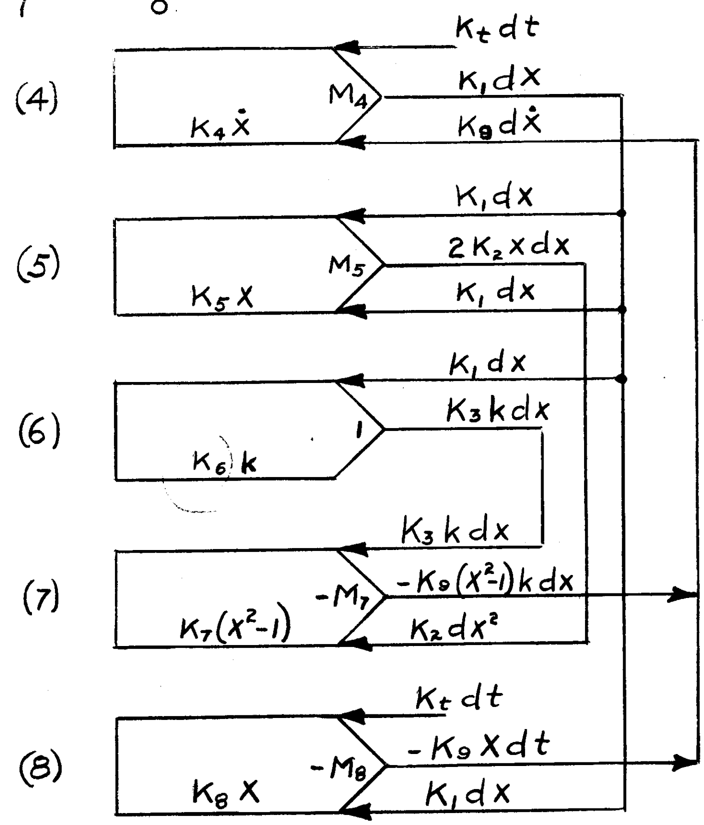


FIGURE 35

There are a large number of sets of scale factors which will satisfy the above relationships. A direct approach to finding a set consists in trying scale values in the map and adjusting the values until a good solution is found. A first solution obtained by this approach is shown in fig. 36.

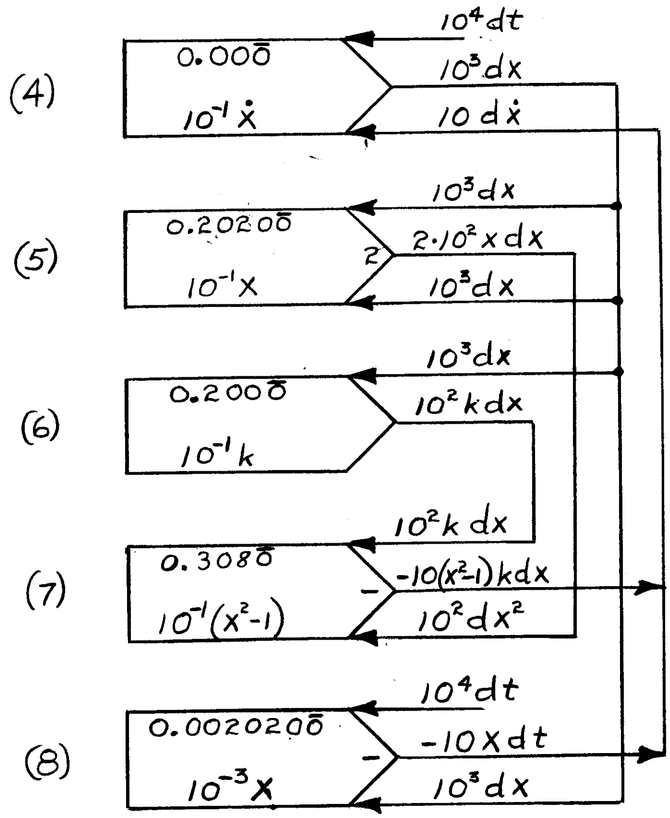


FIGURE 36

The step by step assignments of these scale factors were as follows:

1) K_t was assigned the value of 10^4.

2) M_4 was set equal to 1.

3) K_4 was set equal to 10^{-1} making K_1 equal 10^3, and $M_4\ K_4\ |\dot{x}|$ max about 0.4.

4) M_5 was set equal to 2.

5) K_5 was set equal to 10^{-1} making $M_5 K_5 |X|$ max about 0.4 and K_2, the scale factor of (2XdX), equal 10^2.

6) K_6 was set equal to 10^{-1} making M_6 K_6 k equal 0.2 and K_3 equal 10^2.

7) M_7 was set equal to 1.

8) K_7 was set equal to 10^{-1} making $M_7 K_7 |X^2-1|$ max about 0.3 and K_9 equal 10.

9) M_8 was set equal to 1.

10) K_8 was set equal to 10^{-3} to make the scale factor of the output integrator 8 equal 10.

This first solution is a correct one but it is not an efficient one since all of the $M K_Y |Y_p|$ max values are so much less than 1. The first solution was changed to the more efficient solution shown in fig. 37. The step by step changes from solution number one to solution number two were as follows:

1) M_4 was changed to 2 making K_1 equal $2 \cdot 10^3$ and $M_4 K_4 |\dot{X}|$ max about 0.8.

2) K_5 was changed to $2 \cdot 10^{-1}$ to make $k_5 = k_1$, increasing $M_5 K_5 |X|$ max to about 0.8 and K_2 to $4 \cdot 10^2$.

3) K_6 was changed to $5 \cdot 10^{-1}$ increasing $M_6 K_6$ k to 1 and K_3 to 10^3.

4) K_7 was changed to $4 \cdot 10^2$ to make $k_7 = k_2$

5) M_7 was changed to 5 making $M_7 K_7 |X^2-1|$ max about 0.6 and making K_9 equal $2 \cdot 10^2$.

6) K_8 was changed to $2 \cdot 10^{-2}$ to make the scale of the output of integrator eight equal $2 \cdot 10^2$ as well as make $k_8 = k_1$.

7) K_4 was changed to $2 \cdot 10^{-1}$ to make $k_4 = k_9$.

8) M_4 was changed back to 1 to keep K_1 equal to $2 \cdot 10^3$.

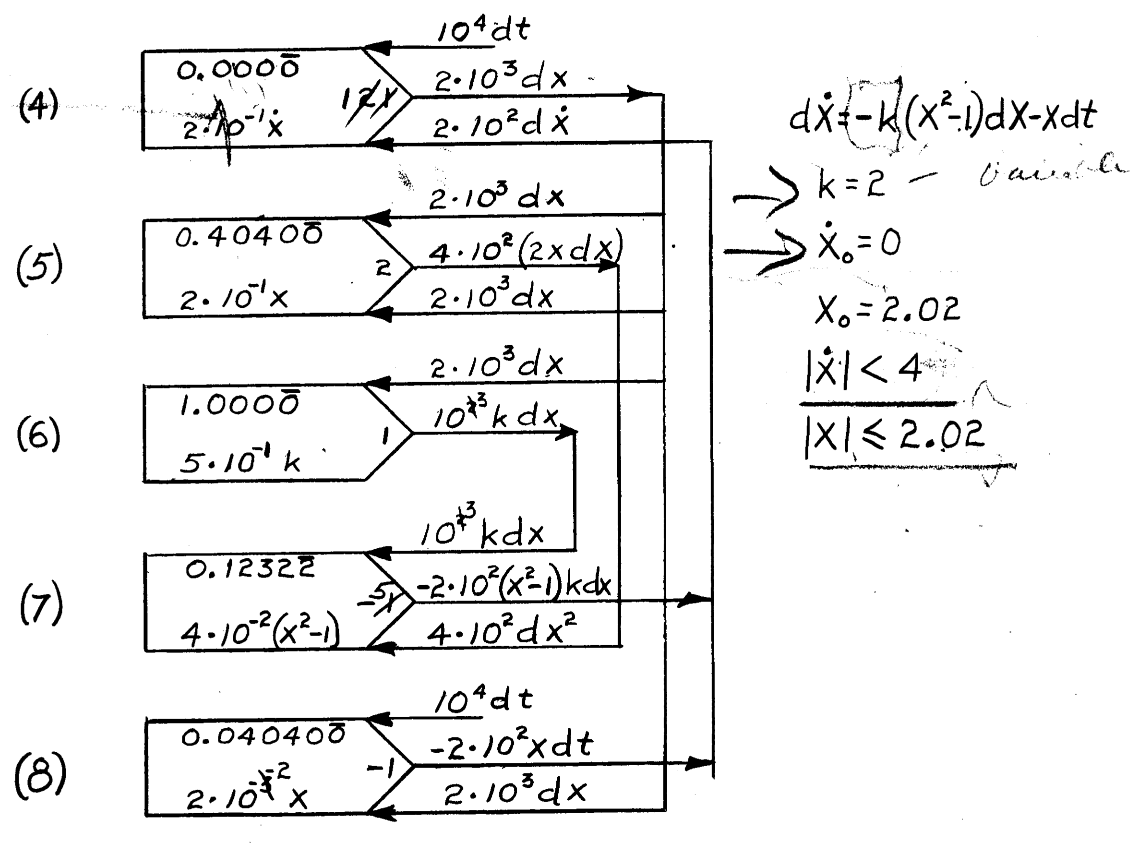


FIGURE 37

The second solution can be adjusted to a third still more efficient solution such as indicated below.

$K_1 = 2.47 \cdot 10^3$	
$K_2 = (2.47)^2 \cdot 10^2$	
$K_3 = 20(2.47)^{-1}\ 10^2$	
$K_4 = 2.47 \cdot 10^{-1}$	$M_4 = 1$
$K_5 = 2.47 \cdot 10^{-1}$	$M_5 = 2$
$K_6 = 20(2.47)^{-2}\ 10^{-1}$	
$K_7 = (2.47)^2\ 10^{-2}$	$M_7 = 5$
$K_8 = 2.47 \cdot 10^{-2}$	$M_8 = 1$
$K_9 = 2.47 \cdot 10^2$	

However, the second scaling solution was used to fill in the scaling terms of the complete map shown in fig. 38.

Integrator 9 was programmed to cause a typeout at 0.1 unit intervals in t. If the desired interval of a variable, u, controlling typeout is Δu and the scale factor of du is Ku, then Y for the typeout control integrator is--

(Eqn. 29) $$Y_T = \frac{1}{(\Delta u)(Ku)} \qquad \frac{1}{.1\ 10^4} = .001$$

The scaling constants, C_1, C_2, C_3 and C_4, in integrators 0, 1, 18 and 19, are such as to make the scale factors on all outputs to the plotters 10^2.

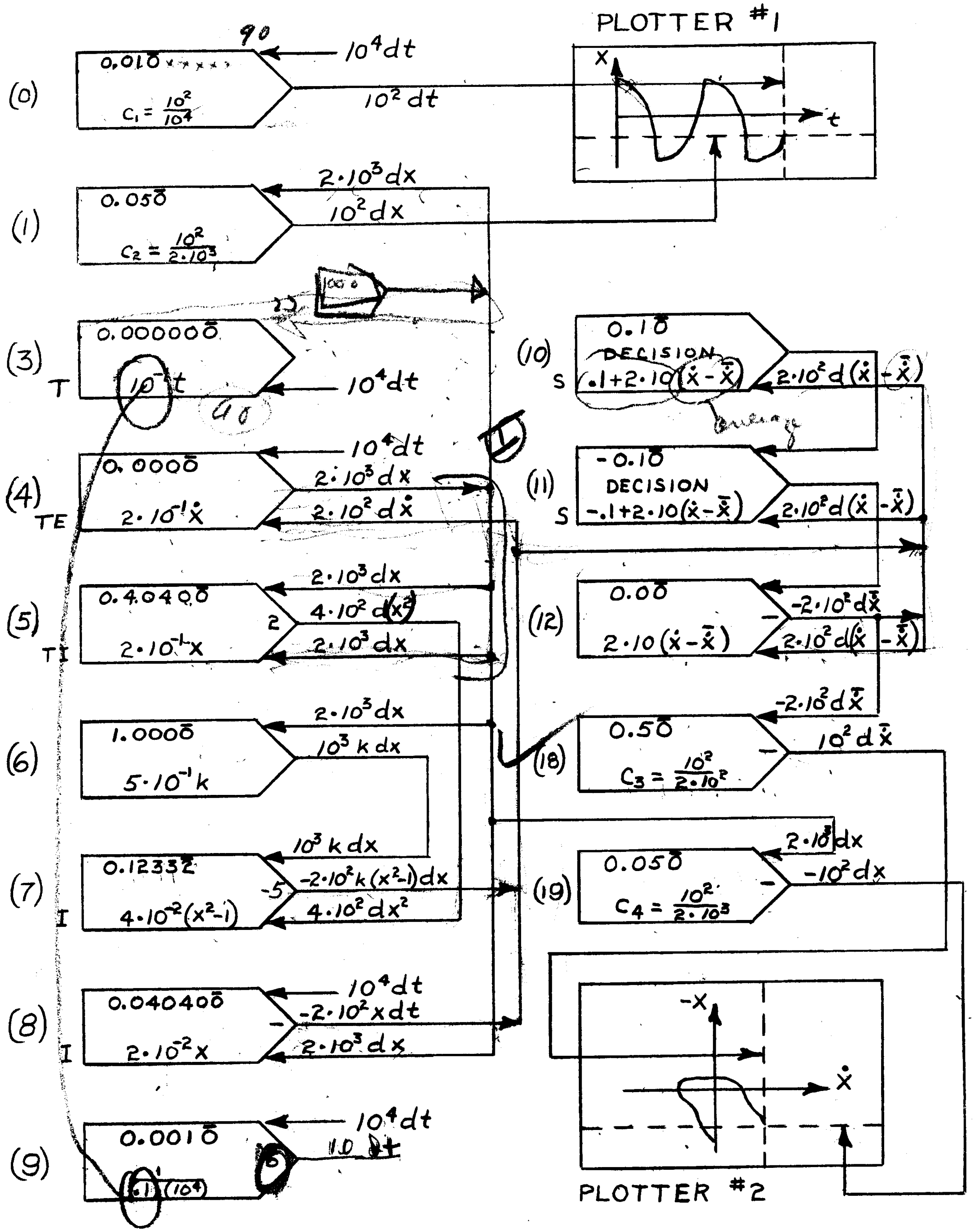

PLOTTER #1
(0)
0.010
$C_1 = \frac{10^2}{10^4}$
$10^4 dt$
$10^2 dt$
(1)
0.050
$C_2 = \frac{10^2}{2\cdot10^3}$
$2\cdot10^3 dx$
$10^2 dx$
(3)
T
0.000000
$10^{-1}t$
$10^4 dt$
(4)
TE
0.0000
$2\cdot10^{-1}\dot{x}$
$10^4 dt$
$2\cdot10^3 dx$
$2\cdot10^2 d\dot{x}$
(5)
TI
0.40400
$2\cdot10^{-1}x$
$2\cdot10^3 dx$
$4\cdot10^2 dx^2$
$2\cdot10^3 dx$
(6)
1.0000
$5\cdot10^{-1}k$
$2\cdot10^3 dx$
$10^3 k\,dx$
(7)
I
0.12332
$4\cdot10^{-2}(x^2-1)$
$10^3 k\,dx$
$-2\cdot10^2 k(x^2-1)dx$
$4\cdot10^2 dx^2$
(8)
I
0.040400
$2\cdot10^{-2}x$
$10^4 dt$
$-2\cdot10^2 x\,dt$
$2\cdot10^3 dx$
(9)
0.0010
$10^4 dt$
(10)
S
0.10
DECISION
$.1+2\cdot10(\dot{x}-\bar{\dot{x}})$
$2\cdot10^2 d(\dot{x}-\bar{\dot{x}})$
(11)
S
-0.10
DECISION
$-.1+2\cdot10(\dot{x}-\bar{\dot{x}})$
$2\cdot10^2 d(\dot{x}-\bar{\dot{x}})$
(12)
0.00
$2\cdot10(\dot{x}-\bar{\dot{x}})$
$-2\cdot10^2 d\bar{\dot{x}}$
$2\cdot10^2 d(\dot{x}-\bar{\dot{x}})$
(18)
0.50
$C_3 = \frac{10^2}{2\cdot10^2}$
$-2\cdot10^2 d\bar{\dot{x}}$
$10^2 d\bar{\dot{x}}$
(19)
0.050
$C_4 = \frac{10^2}{2\cdot10^3}$
$2\cdot10^3 dx$
$-10^2 dx$
PLOTTER #2

FIGURE 38

Since the plotters operate at 100 increments to the inch, the curves will be plotted on a scale of one inch per unit. If P_X is the desired number of inches per unit of X and K_X is the scale factor of X, the plotter scaling constant should be--

$$\text{(Eqn. 30)} \qquad Y_S = \frac{P_X \, 10^2}{K_X} \leqslant 1.$$

The maximum output rate to the plotter should be limited to .4 when 60 integrator operation is used, .2 when 30 integrator operation is used. An upper limit on the output rate of $\dot{X}$ can be found by assuming that the integrands of integrators 4, 6, 7, 8 and 18 are maximum simultaneously. This gives--

$$\left\{ \underset{8}{[.0404]} + \underset{7}{[5\,(.12332)]} \; \underset{6}{[1]} \; \underset{4}{[.8]} \right\} \; \underset{18}{[.5]} = .27$$

Integrators

This indicates that the plotter output rate of $\dot{X}$ may be near, if not greater than, the limit for thirty integrator operation. The other plotter output rates are all within this limit since their output scaling constants are less than .2.

The scale factor on the output of an adder is the same as the scale factor on its inputs. An adder will fail to give a correct answer when the sum of its inputs exceeds a full output rate. For the adder shown in fig. 39 K_A must be chosen so that

(Eqn. 31) $$K_A \leq \frac{K_t}{\left|\frac{du}{dt}\right| \max}$$

or the adder cannot "keep up" with its inputs.

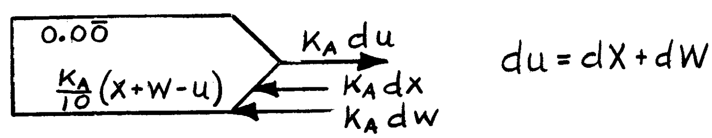


FIGURE 39

A servo, such as shown in fig. 40, is subject to a similar restriction, namely, that--

(Eqn. 32) $$K_S \leq \frac{K_t}{\left|\frac{du}{dt}\right| \max}$$

in order for the servo to "keep up".

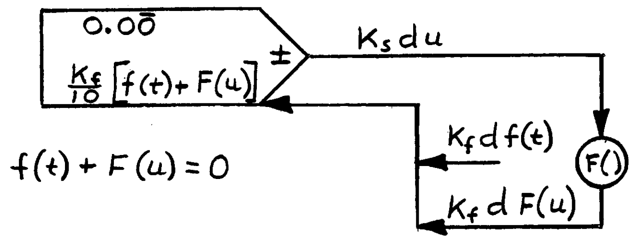


FIGURE 40

The output of a servo can be assigned any value, K_S, which satisfies equation 32 and be used to generate dF(u) at the same scale factor as that for df(t).

The use of the servo program indicated in fig. 41 can be used without restricting the choice of K_S.

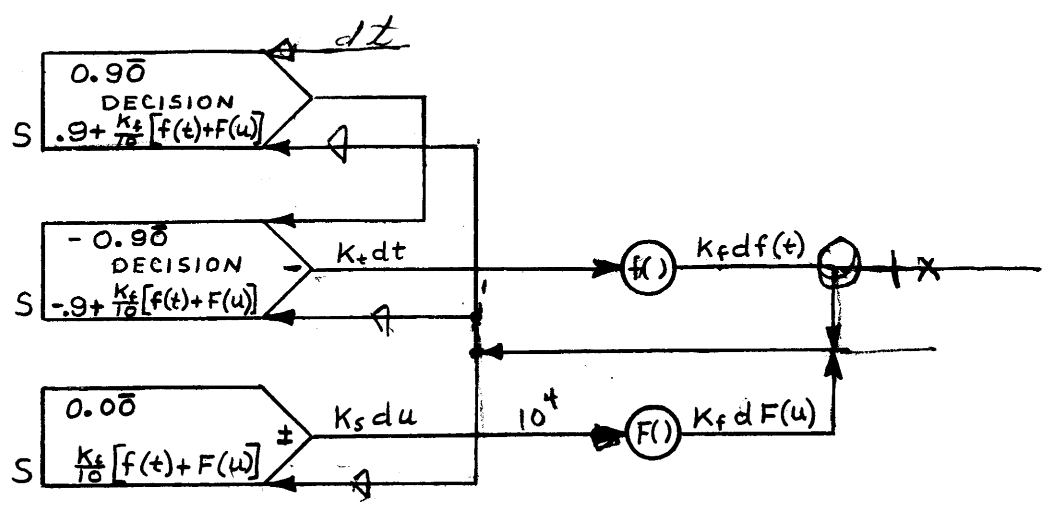


FIGURE 41

Whenever the functions $f(t) + F(u)$ differs from zero by one increment, the independent variable of the problem will not advance until $f(t) + F(u) = 0$. This program will slow up computation only when $K_S \left|\frac{du}{dt}\right| > K_t$.

1.9.3 Coding: The coding for a problem may be put on paper tape using the typewriter to control the tape punch. To code the example problem in fig. 38, the typewriter should be operated so as to obtain a coding sheet such as shown in fig. 42. Each line in this coding sheet represents the coding for one integrator. The coding should be typed as follows:

Correlate [illegible] with preceding information

INT — SGN

```
         11111111112222222222333333333344444444445 *
12345678901234567890123456789012345678901234567890
```

INT							
00	0010	11111	90				
01	0050	11111	04				
03	0000000	11211			90		
04	00000	31211	90		07	08	
05	040400	21212	04		04		
06	10000	11111	04				
07	012332	211-5	06		05		
08	0040400	211-1	90		04		
09	00010	11116	90				
10	010	91111			07	08	12
11	-000	91111	10		07	08	12
12	000	111-1	11		07	08	12
18	050	111-1	12				
19	0050	111-1	04				

FIGURE 42

*Column number designations

1) The first key pressed at the beginning of each line is the carriage return key.

2,3) The number of integrator being coded.

4) Space

5) The initial sign of Y(and Y_I).

(Space) for positive

(-) for negative

6) The initial units digit of Y(and Y_I).

7-14) The initial fractional part of Y(and Y_I) followed by spaces for the unused capacity of the Y registers.

15) Space

16) Integrator operation command.

1 for rectangular integration

2 for interpolative mode

3 for extrapolative mode

4 for multiplicative mode

5 adder

9 servo

17) Integrator reset command

1 for normal

2 for automatic reset

18) Integrator typeout command

1 for normal

2 for integrators to be typed out

19) Integrator output sign reversal.

1 for normal

(-) for output sign reversal

20) Integrator output multiplier

1 for output multiplier of 1

2 for output multiplier of 2

5 for output multiplier of 5

6 for typeout control integrator [This integrator will control]

21) Space

22,23)* The number of the integrator from which the primary input is to be received (if spaces are used, dX = +1 for each iteration). I C.

24) Space

25,26)* The number of the integrator from which the input to the initial condition register is to be received (if spaces are used, $dY_I = 0$ for each iteration). use it only when initial conditions are changed during prob

27, 30, 33, 36, 39, 42, 45, 48) Spaces

28,29; 31,32; 34,35; 37,38; 40,41; 43,44; 46,47; 49,50)* The numbers of integrators from which eight or less secondary inputs are to be received (spaces indicating zero dY inputs).

*(When the machine independent variable is to be an input, "90" is typed and "91" is typed when the negative of the machine independent variable is to be used. These "sources" may not be coded as inputs to an adder).

Carriage return may be typed in place of any space when no more further information is to be typed in the line. This applies to columns 21, 24, 27, 30, 33, 36, 39, 42, 45 and 48 in fig. 42.

Carriage return may be typed at any point in a line where an error was made and the complete line typed correctly. The last line containing coding for a given integrator will be the coding used for that integrator. The integrator numbers in columns 2 and 3 of fig. 42 need not be in numerical order. A period following the last carriage return indicates the termination of the problem coding.

Negative Y values typed on the coding will be filled as "nines complements" resulting in an "error" of 1 in the round-off position. For example, if -0 2 4 3 6 $\overline{2}$ is typed for the Y of an integrator, 9 7 5 6 3 $\overline{7}$ will be filled in the Y register. And 9 7 5 6 3 $\overline{7}$ corresponds to - 0 2 4 3 6 $\overline{3}$ in the Y register. This error is trifling but, if exactly - 0 2 4 3 6 $\overline{2}$ is desired in the Y register, - 0 2 4 3 6 $\overline{1}$ should be typed in the coding sheet. Special integrators often must have exact negative initial conditions. Integrator 11 in the example problem must have an initial Y of - 0 1 $\overline{X}$ (the round-off digit will have no influence on its operation). The coding in fig. 42 shows

- 0 0 0 typed for the initial value of the Y for integrator 11. This will make the initial value of Y, 9 9 $\bar{9}$, or one dY increment negative.

With the exception of a few special cases the entire programs of problems can be entered from properly prepared punched tape. If incremental inputs from sources external to the computer are to be addressed to integrators, these inputs must be addressed from the manual fill keyboard. Also the R registers of some integrators may require manual filling.

OPERATING PROCEDURES

2.1 Power On Procedure

When the computer is connected to a suitable power source (see Maintenance Manual), pressing the black button marked ON in the upper left-hand portion of the control panel (see fig. 43) supplies power to the AC circuits in the computer and lights the red light marked AC. After approximately one and one-half minutes, dc is automatically turned on together with the red light marked DC.

When dc is turned on, it may happen that the various flip-flops controlling the tape punch, both the action and direction of the tape reader, the typewriter, and the overflow lights, are set to the on position. The RESET button resets all of these flip-flops and thus may be used not only when the computer is being turned on but at any time to stop an ongoing process involving the tape units, to stop the typewriter, to set the tape reader to the forward direction, or to turn the overflow lights off.

NOTICE: High voltage is present within the control console, therefore extreme care should be observed in removing any of the access panels.

when putting in a new roll of tape procedure used.

The path to be followed in threading the new tape is as follows: over the roller, through the tape guide, through the rear opening in the cover, down the tape guide on the punch, through the slot in the punch block, between the tape feed roller and its tape feed pressure clip, through the slot guide in the front of the punch, and through the front opening in the cover.

When a small length of tape protrudes from the slot guide in front of the punch, a slight tension may be applied to the end of the tape, the PUNCH ON lever switch depressed and the tape RUN button pressed for a few seconds to bring the punched holes in the tape into registry with the tape feed roller.

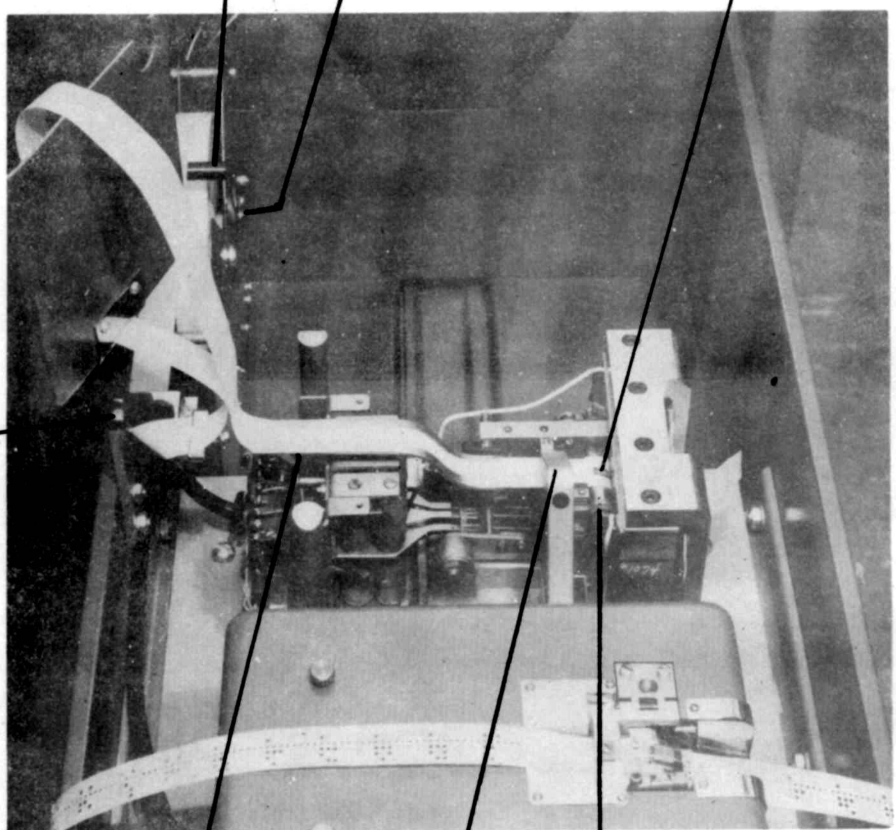
2.3 Reading of Tape

To set the tape into the reader, the TAPE LATCH must first be twisted so as to allow the TAPE GATE to swing up. The tape is now placed so that one of the tape sprocket holes in the leader portion of the tape is on one of the TAPE SPROCKET teeth and the tape is centered on the guide ridges. With the tape in this position, the TAPE GATE is lowered and the TAPE LATCH twisted so as to hold the GATE down. The tape drive is set to forward by pressing the RESET button.

2.2 Preparation of Problem Tapes

A tape may now be punched according to the instructions for problem coding given in 1.9.3. The tape equipment is located in the lower left-hand drawer and is shown in fig. 44. The punch is made operative by depressing the lever switch marked PUNCH ON on the typewriter (fig. 45). A blank section of tape preceding the actual problem information must now be produced in order to facilitate handling and marking of the completed tape. This "leader" is provided by pressing the Tape RUN Button, located underneath the typewriter. Five seconds is sufficient to provide a six inch leader. The typewriter power switch is now turned to ON and the information for the problem typed. After the last character another blank section of tape is produced, the completed tape is torn off, and the PUNCH ON switch is elevated.

2.2.1 Replenishing Tape Supply: When a new roll of tape is to be installed, the old tape core and the hub are removed from the tape container, shown in fig. 44a, and the remaining tape is run or pulled out of the punch. The hub is put in the new roll of tape and is dropped into position in the tape container. In order to thread the tape through the punch, the cover on the punch must be removed by loosening the four thumb screws on top and lifting it off.


ROLLER
TAPE CONTAINER
TAPE FEED PRESSURE CLIP
TAPE GUIDE
TAPE GUIDE
PUNCH BLOCK
FEED ROLLER

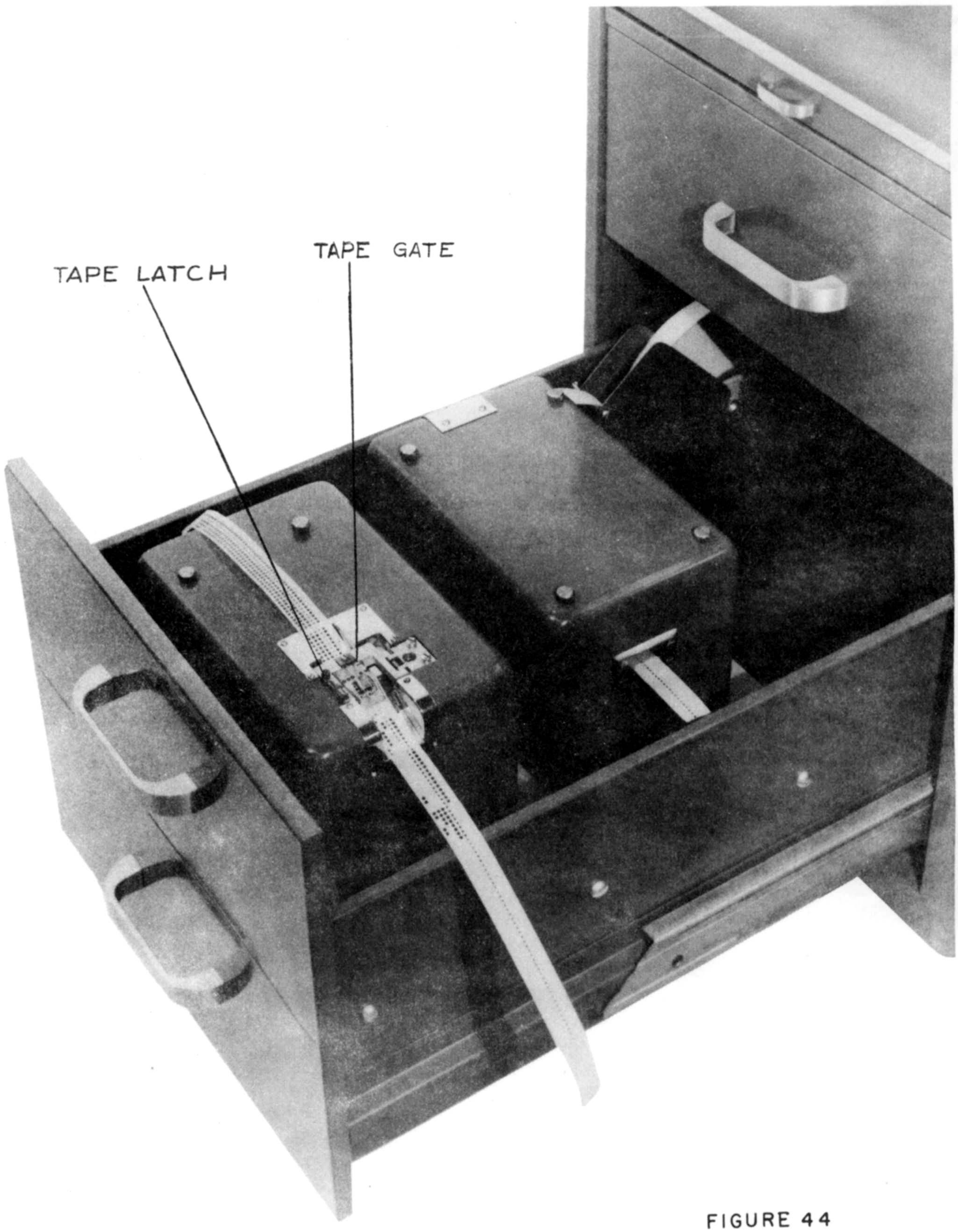


FIGURE 44

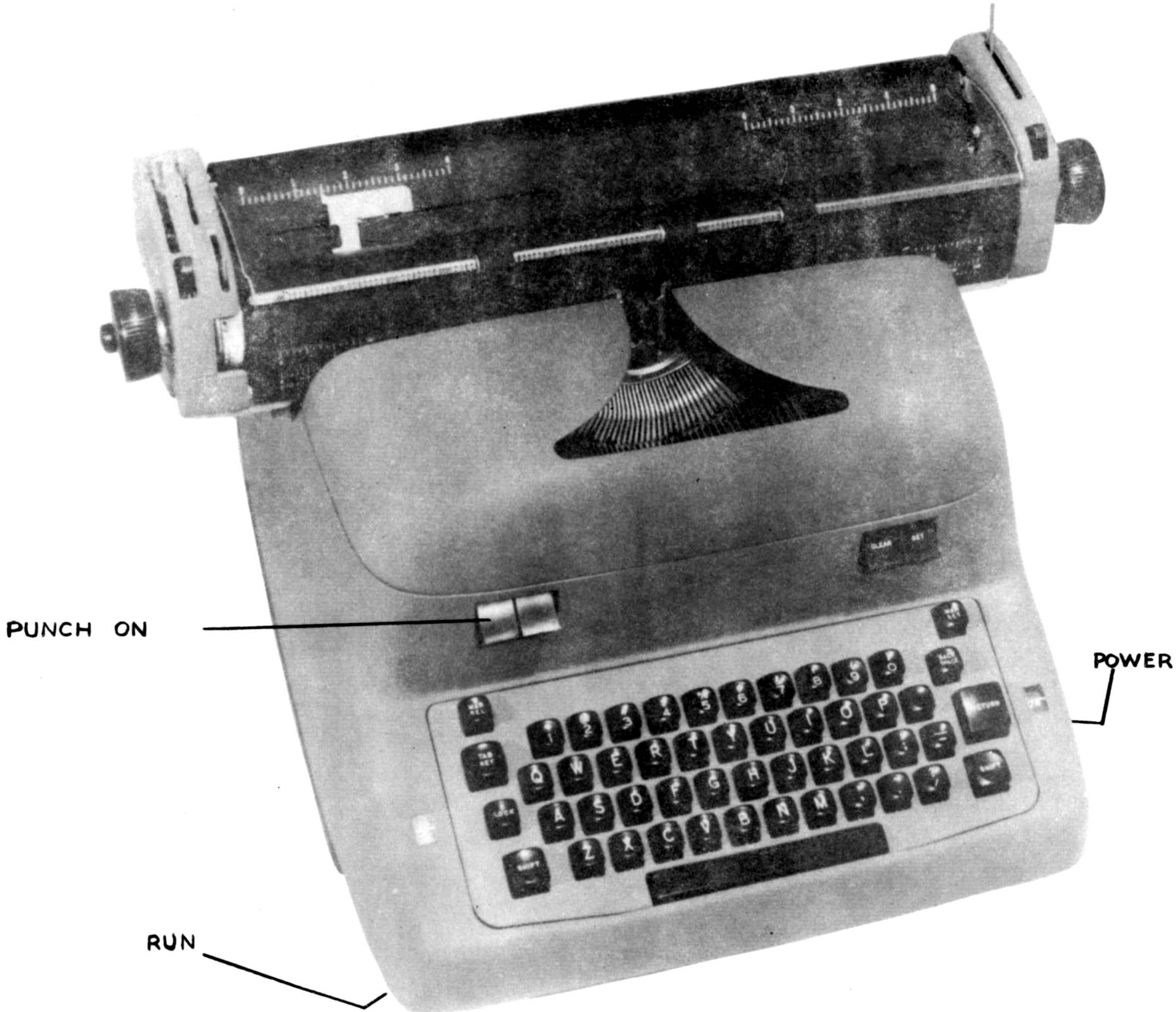


FIGURE 45

Once the tape is in position, the computer must be cleared of any old information by depressing the black CLEAR button marked COMPLETE in the center of the control panel. The FILL-IDLE-RUN toggle is set to FILL and the READ-IN switch to F. It is important to set the 30-60 switch appropriately at this point. Finally, the TIME switch must be set to the center position. The Fill process may now be initiated by pressing the green button in the center of the READ-IN switch. As information is read from the tape it may be typed by the typewriter if the typewriter power switch is in the ON position; and, in addition, if the PUNCH ON switch is depressed, a duplicate tape will be produced. That is, it is possible to produce a typewritten transcription of the tape without reproducing the tape but it is not possible to reproduce the tape without typing the information.

2.4 Computation Involving Neither Input Nor Output Functions

When the filling tape stops at its period code, the problem is in the computer, the integrands being held in the Y_1 registers only. If the punch happens to be on, it should be turned off unless a tape copy of the answer is desired. The READ-OUT switch is next set to N, the READ-IN to N and the TIME switch to the appropriate position. Finally, to insert the integrands into the Y and Y_D registers, the INSERT button is pressed. The computer is now ready to compute the solution of the problem that has been entered. At this point a sheet of paper is inserted into the typewriter in order to record the answers. The initial condition of the variables to be printed will be typed out if the green button in the CENTER of the READ OUT switch is pressed. The FILL-IDLE-RUN switch may now be set to RUN and computation started. The RUN light indicates on-going computation.

2.5 Use of the Graph Plotter

The graph plotter unit is shown in fig. 46. If the problem being run is coded to use the plotter, the Power Switch is set to ON, lighting the Power Light. By use of the RELEASE X and RELEASE Y buttons, the drum and stylus holder, respectively, may be made to move freely facilitating both the attaching of paper to the drum and its rough positioning. The Cross Hair cursor in conjunction with the incremental movement buttons provide for precise positioning after which the stylus itself is inserted. It is important that neither the drum nor the holder be moved without first pressing the appropriate RELEASE. If the plotter is not being used it should be turned off.

2.6 Display

The arrangement of information in the memory is shown in fig. 47,

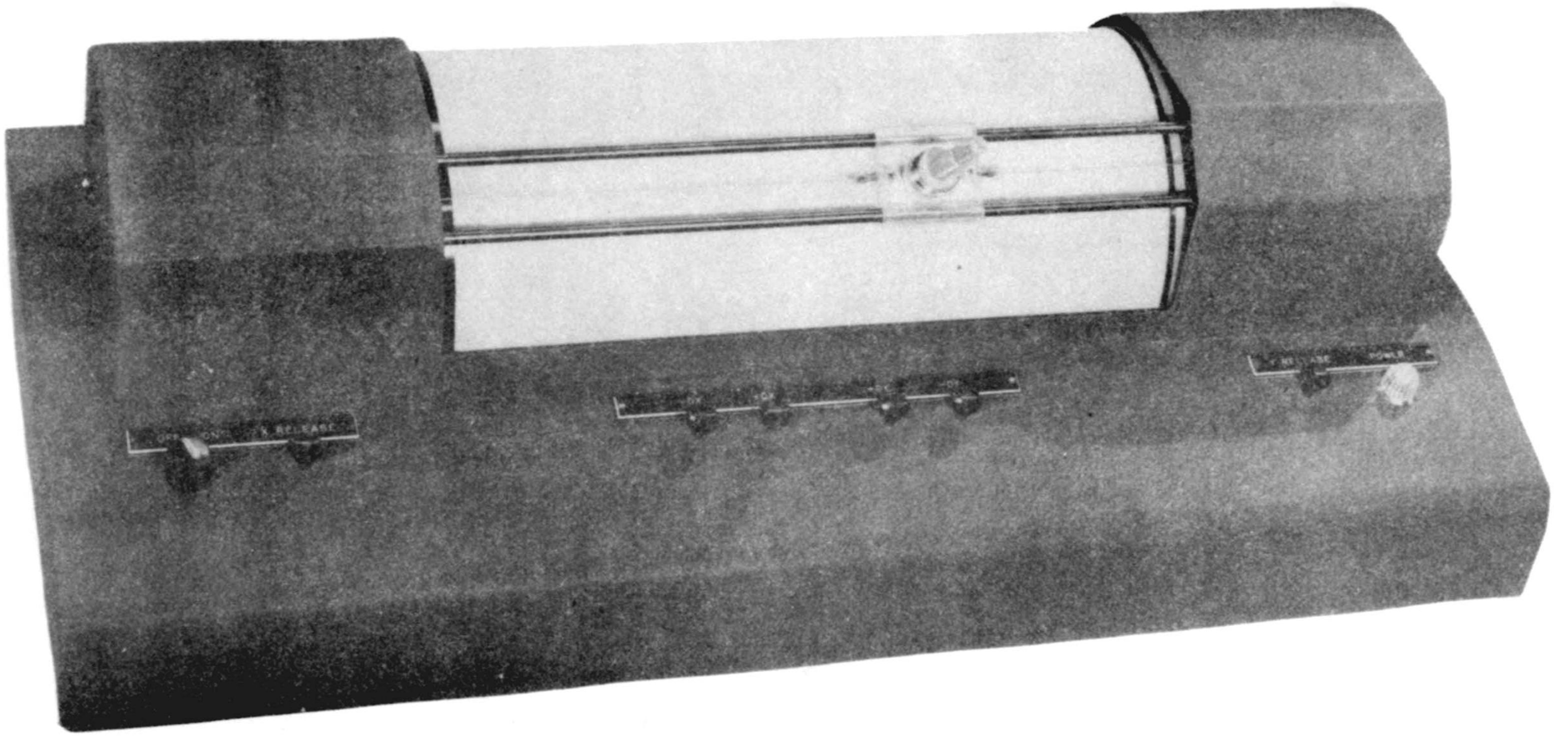


FIGURE 46

(Address of the integrator) This chart is to give information about the integrator so that it can identify itself

Arithmetic register: tells actual information to be processed on the integrator.

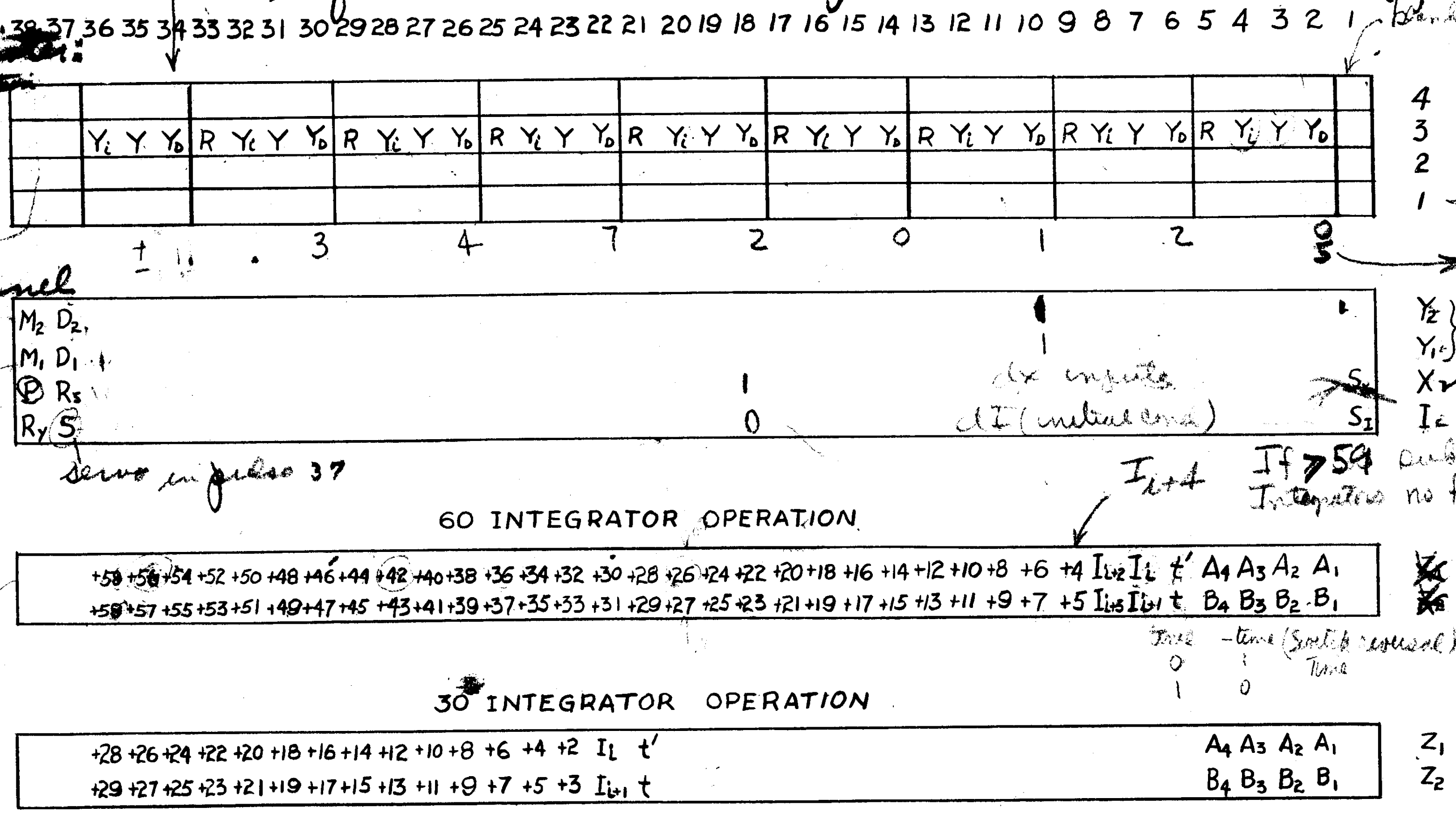


FIGURE 47

$M_2' M_1'$: MULTIPLIER EQUAL TO 1
$M_2' M_1$: MULTIPLIER EQUAL TO 2
$M_2 M_1'$: MULTIPLIER EQUAL TO 5
$M_2 M_1$: TYPEOUT CONTROL

$D_2' D_1'$: RECTANGULAR MODE
$D_2' D_1$: EXTRAPOLATIVE MODE
$D_2 D_1'$: INTERPOLATIVE MODE
$D_2 D_1$: MULTIPLIER MODE

P : TYPEOUT SELECTOR
R_y: SELECTIVE RESET SELECTOR
S : SERVO
R_S: SIGN REVERSAL
S_X: READ DX IN Z_2
S_I: READ DI IN Z_2

DECIMAL NUMBER	0	1	2	3	4	5	6	7	8	9
EXCESS 3 BINARY CODE	0	0	0	0	0	1	1	1	1	1
	0	1	1	1	1	0	0	0	0	1
	1	0	0	1	1	0	0	1	1	0
	1	0	1	0	1	0	1	0	1	0

primed are zeros
unprimed are ones

the numbers at the top being timing numbers. The integrator from which information is being displayed by the oscilloscope is normally controlled by the INTEGRATOR switches; the dial on the left holds the most significant digit, that on the right the least significant digit. However, during the processing of either FILL information or of monotonic functions, the information being processed controls the display.

The portion of the integrator that is displayed is controlled by the DISPLAY switch. When in either the I, Y, D, or R positions, the portions of the 4, 3, 2, and 1 lines which hold Y_i, Y, Y_D, and R respectively, are displayed. A bright spot represents a binary one in the coded binary scheme shown in fig.47.

If the switch is in either the ΔI, ΔX, Δ1, or Δ2 positions, the complete address lines I, X, Y_1 and Y_2 corresponding to that integrator are shown. Note that with respect to the integrator timing system, this is I_{i-1} if the selected integrator is I_i. Control information is placed in times 37 and 38, addresses in positions 1 through 36.

Two channels, Z_2 and Z_1 hold the outputs of the integrators as indicated in fig.47. A one in any of the address positions indicates that the information in the corresponding Z line is to be used as an input. Two Y lines are used corresponding to the two Z lines. However, since the dX and dY_I inputs are limited to at most one each, only one X and one I address line is required. The S_X and S_I digits in position 1 indicate which of the Z lines is to be interrogated, a one indicating that the addresses are to be considered with respect to Z_2.

It should be noted that the arrangement of integrators on the Z lines shown in fig.47 is relative, that is, the number of the integrator being filled must be employed in determining the position of any integrator output with the exception of positions 2 through 6. The latter are identical in all integrators--holding the eight possible incremental inputs in positions 2 through 5 and plus and minus time in position 6.

2.7 Overflow and Proscribed Code Indication

Two amber lights in the upper right-hand portion of the control panel indicate the presence of an overflow and a proscribed code in the computer. The light marked OVERFLOW is turned on whenever an integrator attempts to create an output of +2 or -2 while the CODE light is turned on in the event that a code which does not represent a decimal number appears anywhere in the arithmetic channels.

In addition, position one of Y_2 is filled in the second integrator after the integrator in which the overflow or proscribed code occurred. Pressing the RESET button turns the lights off, however, the one that has been inserted must be removed manually. (see section 2.8)

2.8 Manual Fill

Normally, the computer is filled automatically. It may happen, however, that a mistake in coding or in the preparation of a tape necessitates one or several changes in the information held in the computer. To implement such changes, facilities are provided for the manual determination of each bit of the address and arithmetic information. First the DISPLAY and INTEGRATOR switches are set to the proper positions, the READ-IN and READ-OUT switches set off G, the FILL-IDLE-RUN toggle set to FILL and the MARKER bar to the right of the decimal keyboard on the control panel is pressed. This inserts a pulse in the T channel at P_{36} of the selected integrator if arithmetic information is to be changed or of the immediately preceding integrator if an address is to be modified.

In the former case no indication of the positioning of the marker is presented, but it is automatically moved to the next position on the right each time a decimal key, the NULL bar or the → bar directly above the decimal keyboard is pressed. In order to change any number, the → bar is pressed once for each digit to the left of the position to be changed and the new digit entered. For example, if Y_I is to be changed from 0.76942 to 0.77942, the bar is depressed twice after which a 7 is entered. The NULL bar provides for the erasure of digits in that it sets all the binary digits in the selected position to zero.

If an address position is to be changed, the position of the marker is indicated by a bright spot (a one) in all four address channels and moves only under the control of the ← and → bars. The marker is positioned to the desired pulse time (see section 2.6) by these right and left direction bars, and the 1 key pressed if an address is to be entered, the 0 bar is an erasure is to be made.

It may happen that a complete integrator is to be removed from the program or that a change is facilitated if part of the integrator in question is initially cleared. The CLEAR buttons marked VARIABLE and ADDRESS clear the arithmetic registers (I_I) and the address channels (I_{I-1}) respectively, of the integrator selected by the INTEGRATOR switch.

After a change has been made, or at any other time, the value of the integrand in the integrator selected by the INTEGRATOR switch may be typed out by pressing the green button in the center of the most significant digit part of the switch, providing the FILL-IDLE-COMPUTE toggle is not in the FILL position.

2.9 Use of Non-Incremental Input Functions

2.9.1 Manual Preparation of Tapes for Monotonic Functions:

The following chart shows the information that must be typed after the punch has been turned on and a leader produced.

Period

CR (carriage return)

Integ. Number (2 digits)

Space

Sign and Value of $I_1(1)$

CR

Integ. Number

Space

Sign and Value of $I_2(1)$

.
.
.
.

CR

Integ. Number

Space

Sign and Value of $I_n(1)$

CR

Period

} 1st point of independent variable

Cr

Integ. Number

Space

Sign and Value of $I_1(2)$ — 2nd point of independent variable

CR

etc. — etc.

The sign should be typed as a space for positive and "-" for negative values of the input functions. The values of the input functions may be expressed with up to nine digits, the first being the units digit and the last being the round-off digit.

2.9.2 Manual Preparation for Reversible Tapes: The number of reversible functions is limited to one. The following chart indicates the information put on the tape after the leader,

CR

4 spaces

Value of I(1)

CR

4 spaces

Value of I(2)

CR

4 spaces

Value of I(3)

etc--

The values may be typed with four to eight digits and must be expressed as complements for negative values.

2.9.3 Auto Preparation of Reversible Tapes: In some cases a function generated during the solution of one problem can be employed as an input function to another problem. To facilitate this, the computer can punch a tape with the required results of the first problem in a proper form for use in the second problem.

To prepare a tape of this type, the usual procedures are followed except that the READ-OUT switch is set to R. The typewriter must be on and the PUNCH ON switch lowered. A leader of tape should be obtained by pressing the TAPE RUN button after which a carriage return and a space is typed. Computation may then be started by setting the FILL-IDLE-RUN toggle to RUN. The one integrator coded for typeout will be tabulated and a reversible tape punched automatically.

It should be noted that a tape may be punched which results from computations involving function inputs. This is possible since the read-in and read-out operations are independent.

2.9.4 Computation With Input Functions: When input functions are employed, the procedure described in IV is followed in loading information into the computer except that before pressing the INSERT button the function tape is entered into the tape reader. (Pressing the RESET button assures that the tape drive is in the forward position.)

2.9.4.1 Monotonic Function Inputs: The tape is entered into the reader at the period preceding the input values of the first point to be used. The READ-IN switch is set to M and the INSERT button pressed. This automatically reads the values for the first point into the Y_I registers of the appropriate integrators. After these values have been read from the tape, the INSERT button is pressed once more to insert all the initial conditions of Y into the Y and Y_D registers and to read the values for the second point into the appropriate Y_I registers. The FILL-IDLE-RUN switch may now be set to RUN and the answer computed.

2.9.4.2 Reversible Function Inputs: The tape is set at the CR preceding the value of the initial condition of the integrand of integrator 02. It is now necessary to set the DIGITS switch. Access is afforded to this switch by removing the back panel of the desk. Fig.48 shows the position of the switch. The number of digits to the right

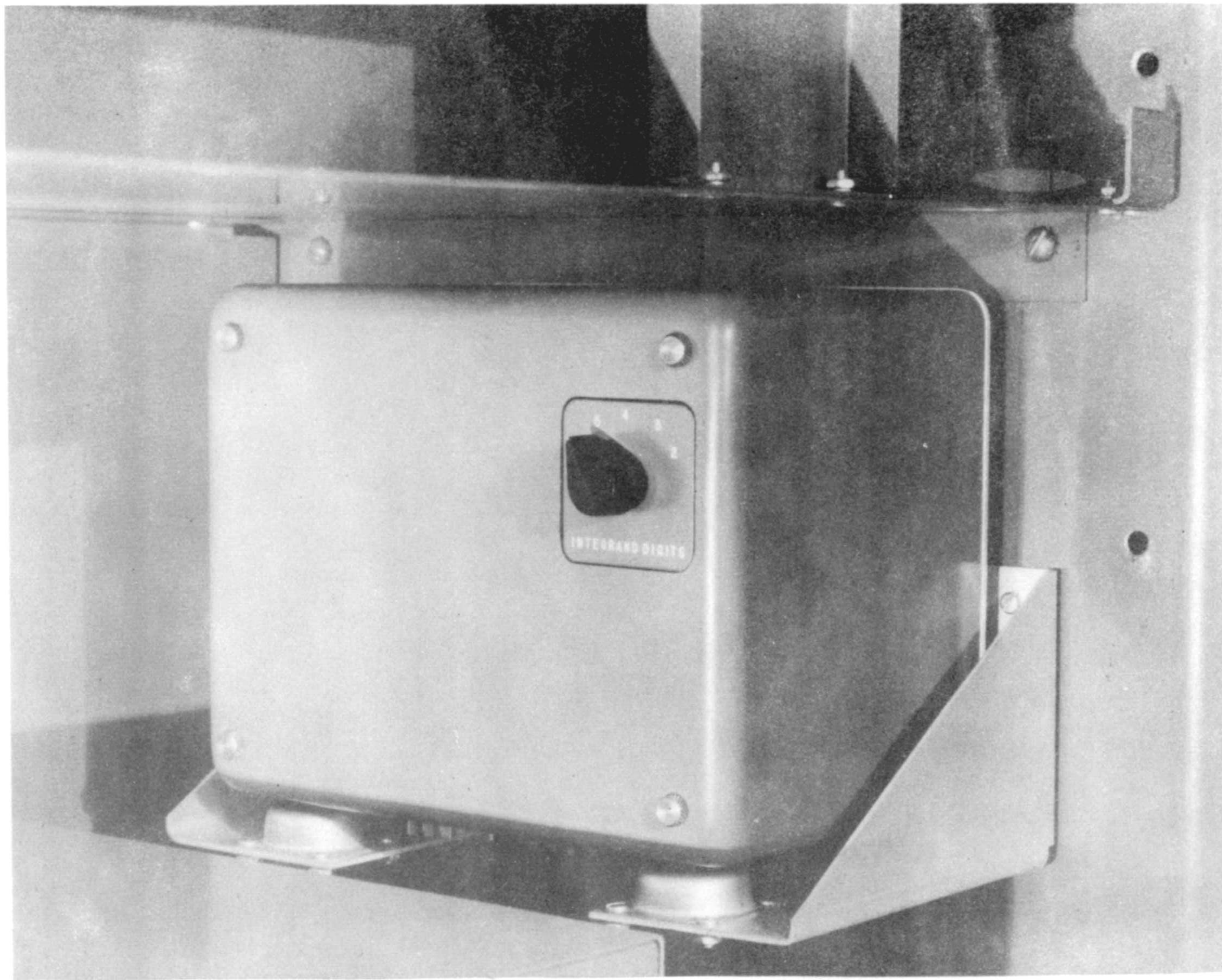


FIGURE 48

of the decimal point in integrator 02 (ignoring the round-off digit) is set into the switch, and the panel replaced. The READ-IN switch is then set to R and the INSERT button is pressed. After the initial condition of integrator 02 has been read into Y_I, the INSERT button is again pressed to insert all the initial conditions of Y into the Y and Y_D registers and to read the value for the second point into the Y_i register of 02. The FILL-IDLE-RUN switch may now be set to RUN and the answer computed. If the computed independent variable reverses its direction of change, the tape feed will be reversed automatically.

2.10 One Cycle Operation

In some cases it is desirable to be able to control computation so that it proceeds one cycle at a time. This may be accomplished by following an appropriate procedure described above with the exception that the last step is changed to setting the FILL-IDLE-RUN switch to IDLE rather than RUN. One cycle will now be computed each time the 0 bar of the decimal keyboard is depressed. Automatic computation may be initiated at any point by switching from IDLE to RUN.

2.11 Problem Read-Out

For purposes of reading out, the computer is divided into two parts, the Arithmetic and Z-line information in one and the Address in another. These may be processed in any order and for any number of integrators. First, the DISPLAY switch is set to either an address or arithmetic position and the INTEGRATOR switch to the lowest numbered integrator to be copied. The READ-OUT switch is set to G and the READ-IN switch to N. After setting the FILL-IDLE-RUN toggle to FILL, the MARKER bar is depressed inserting a marker to signify the starting point of the operation after which the INTEGRATOR switch is reset to the last integrator to be copied. After turning the punch on, the green button in the center of the READ-OUT switch is pressed, starting the operation. It should be noted that reading out does not disturb any of the information held in the computer.

If all sixty integrators are to be copied, approximately 6.5 minutes are required, a smaller number of integrators requiring proportionately less time. After half of the channels have been processed, the DISPLAY switch is set to the other region and the above procedure repeated.

2.12 Problem Read-In

To read the information from the tape described in section 2.11 into the computer, the DISPLAY switch is set to an address or an arithmetic position depending upon the information on the tape. The INTEGRATOR switch is then set to the lowest numbered integrator to be filled. It should be noted that this need not be the same integrator which initiated the read-out procedure since a relative shift of all integrators will not necessarily affect the computation process. A shift of this type is not possible, however, if integrators with specialized functions, such as integrator 59 which controls initial condition reset, and integrators 00 and 01 which feed a graph plotter, are employed in their specialized capacity. After setting the Read-Out switch to N, the READ-IN switch to G, and the FILL-IDLE-RUN toggle to FILL, the MARKER bar is pressed and the INTEGRATOR switch reset to the last integrator to receive information. Again the green button in the center of the READ-OUT switch is pressed to start the process.

If, because of the wearing of a tape or for any other reason a duplicate copy of a problem tape is needed, it may be produced simultaneously with the READ-IN operation. The normal READ-IN instructions are followed with the exception that the READ-OUT SWITCH IS SET TO G and the tape punch is turned on.

2.13 Power Off Procedure

The computer is turned off by pressing the red button in the upper left-hand portion of the control panel marked OFF. When pressed, both the AC and DC lights are turned off and all voltages are removed from the computer proper. The cooling system fans, however, remain on for approximately four minutes.