A PROCESS FOR THE STEP-BY-STEP INTEGRATION OF DIFFERENTIAL EQUATIONS IN AN AUTOMATIC DIGITAL COMPUTING MACHINE

By S. GILL

Communicated by M. V. Wilkes

Received 3 June 1950

ABSTRACT. It is advantageous in automatic computers to employ methods of integration which do not require preceding function values to be known. From a general theory given by Kutta, one such process is chosen giving fourth-order accuracy and requiring the minimum number of storage registers. It is developed into a form which gives the highest attainable accuracy and can be carried out by comparatively few instructions. The errors are studied and a simple example is given.

1. *Introduction.* Most of the methods employed in hand computing for the step-by-step integration of differential equations have an essential feature in common, namely, that at each step of the integration, use is made of the function-values already obtained in preceding steps. Thus if we have arrived at the value of y_n and we wish to find y_{n+1}, these methods require a knowledge of $y_{n-1}, y_{n-2}, \ldots$, the number of values depending upon the method and the accuracy required. Such processes have obvious advantages. The values of $y_{n-1}, y_{n-2}, \ldots$ carry information about the behaviour of y in the region of y_n, and this can assist in determining y_{n+1}; to disregard it would be wasteful. Also these processes, nearly all of which are based on difference formulae, have simple analytical forms and are easily remembered. These points weigh heavily in hand work, where every scrap of information is obtained at the cost of some mental labour, and simple formulae are definitely preferred. There are disadvantages, though not serious ones from the hand computer's point of view. First, these processes cannot be used at the beginning of a range of integration. Two or more consecutive values of y are needed before the main process can operate, and to obtain these some auxiliary process must be used. Secondly, it is not a simple matter to change the size of the interval in the middle of a run. Doubling the interval is fairly easy, halving it is not too difficult, but to change it by any other factor is quite complicated.

When high-speed automatic machines are used, these considerations assume altogether different proportions. It becomes a serious drawback to have to supply the machine with special instructions for a starting process, or for changing the size of the interval. Moreover, a process which appears simple to a hand computer may involve a considerable number of operations in an automatic machine. For example, having found y_{n+1} from, say, y_n, y_{n-1} and y_{n-2}, we then wish by a repetition of the same operations to form y_{n+2} from y_{n+1}, y_n and y_{n-1}. Before this can be done we must replace y_{n-2} by y_{n-1}, y_{n-1} by y_n and y_n by y_{n+1}. These shifting operations consume a serious proportion of the time and the instructions in a machine; the manual computer performs them merely by moving his eyes down the page.

The number of storage registers available in any machine is limited, and if more than twenty or thirty simultaneous equations are being solved the shortage of registers may be serious. For n first-order equations, the number of registers required by any process is of the form $An+B$, which if n is large depends primarily on A. If several consecutive values, or backward differences, of each variable are to be stored at any moment, A will be correspondingly large.

We are thus led to consider, for applications to automatic machines, processes which do not make use of preceding function values. The general theory of a large class of such processes has been given by Kutta (1). Kutta investigated processes of various orders of accuracy; the most attractive are those of the fourth order, in which the error in each step is of the order of h^5, where h is the length of the interval. One of these, known as the Runge-Kutta process, is sometimes used by hand computers for starting an integration. It will be shown that the general case requires $A = 4$, but that certain cases exist which can be carried out with $A = 3$, and the choice and development of one of the latter will be described.

2. *Kutta's fourth-order processes.* Consider first a single first-order equation

$$\frac{dy}{dx} = f(x, y). \tag{1}$$

Suppose that we have arrived at the point $x = X$, $y = Y$, and we want to obtain the value of y corresponding to $x = X+h$. By substitution in (1), we obtain the value of dy/dx at the beginning of the interval; this gives us a first approximation to the curve, as a straight line between (X, Y) and $(X+h, Y+k_0)$, where

$$k_0 = hf(X, Y). \tag{2}$$

We now travel a fraction m of the way along this line and perform another substitution into (1) to obtain

$$k_1 = hf(X+mh, Y+mk_0). \tag{3}$$

The estimates k_0 and k_1 are now combined to find a third point at which f is calculated,

$$k_2 = hf(X+nh, Y+[n-r]\,k_0+rk_1), \tag{4}$$

and finally

$$k_3 = hf(X+ph, Y+[p-s-t]\,k_0+sk_1+tk_2). \tag{5}$$

The four estimates of k are now suitably combined to give the value adopted as the increment of y:

$$\delta y = y(X+h)-y(X) = ak_0+bk_1+ck_2+dk_3, \tag{6}$$

where

$$a+b+c+d = 1. \tag{7}$$

By choosing the coefficients appropriately, the result may be made correct as far as the terms in h^4.* Kutta derived the necessary conditions and suggested several possible solutions.

Before examining these conditions, let us consider the extension to simultaneous first-order differential equations (and thus to equations of higher order). Suppose we have n equations

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_n) \quad (i = 1, \ldots, n). \tag{8}$$

* The process may be extended to give higher orders of accuracy, by introducing more stages into the step, i.e. more quantities in the series k_0, k_1, k_2, k_3. However, fifth-order accuracy requires not five but six stages, and the algebra is considerably more complicated than in the fourth-order case.

The process now takes the following form:

$$\left.\begin{aligned} k_{i0} &= hf_i(X, Y_1, Y_2, \ldots), \\ k_{i1} &= hf_i(X+mh, Y_1+mk_{10}, Y_2+mk_{20}, \ldots), \\ k_{i2} &= hf_i(X+nh, Y_1+[n-r]k_{10}+rk_{11}, Y_2+[n-r]k_{20}+rk_{21}, \ldots), \\ k_{i3} &= hf_i(X+ph, Y_1+[p-s-t]k_{10}+sk_{11}+tk_{12}, \ldots), \\ \delta y_i &= ak_{i0}+bk_{i1}+ck_{i2}+dk_{i3}, \end{aligned}\right\} \quad (i = 1, \ldots, n). \tag{9}$$

It is possible to simplify the theory, and also the form of the process in an automatic machine, by writing $x = y_0$. We now have $(n+1)$ equations

$$\frac{dy_i}{dx} = f_i(y_0, y_1, \ldots, y_n) \quad (i = 0, \ldots, n), \tag{10}$$

where

$$f_0 \equiv 1. \tag{11}$$

With this simplification we have in place of (9):

$$\left.\begin{aligned} k_{i0} &= hf_i(Y_0, Y_1, \ldots), \\ k_{i1} &= hf_i(Y_0+mk_{00}, Y_1+mk_{10}, \ldots), \\ k_{i2} &= hf_i(Y_0+[n-r]k_{00}+rk_{01}, Y_1+[n-r]k_{10}+rk_{11}, \ldots), \\ k_{i3} &= hf_i(Y_0+[p-s-t]k_{00}+sk_{01}+tk_{02}, \ldots), \\ \delta y_i &= ak_{i0}+bk_{i1}+ck_{i2}+dk_{i3}, \end{aligned}\right\} \quad (i = 0, \ldots, n). \tag{12}$$

Now $y_i(X+h)-y_i(X)$ can be expanded as a power series in h, and so can δy_i as given by (12). By equating the terms up to the fourth order in h, we obtain the conditions to be satisfied by the coefficients in (12). Columns 2 and 3 of Table 1 show the results of these expansions, including the terms in h^5 which we require later for an estimation of the error. We thus arrive at the conditions:

Table 1

Term $\left(f_i^j = \left(\frac{\partial f_i}{\partial y_j}\right)_X\right)$	Coefficient in $y_i(X+h)-y_i(X)$	Coefficient in δy_i from (12)	Coefficient in δy_i from (16)	Error in δy_i from (16)
hf_i	1	$a+b+c+d$	1	0
$h^2f_jf_i^j$	$\frac{1}{2}$	$bm+cn+dp$	$\frac{1}{2}$	0
$h^3f_jf_kf_i^{jk}$	$\frac{1}{6}$	$\frac{1}{2}(bm^2+cn^2+dp^2)$	$\frac{1}{6}$	0
$h^3f_jf_k^jf_i^k$	$\frac{1}{6}$	$crm+d(sm+tn)$	$\frac{1}{6}$	0
$h^4f_jf_kf_lf_i^{jkl}$	$\frac{1}{24}$	$\frac{1}{6}(bm^3+cn^3+dp^3)$	$\frac{1}{24}$	0
$h^4f_jf_kf_l^kf_i^l$	$\frac{1}{8}$	$crmn+d(sm+tn)p$	$\frac{1}{8}$	0
$h^4f_jf_kf_l^jf_i^k$	$\frac{1}{24}$	$\frac{1}{2}\{crm^2+d(sm^2+tn^2)\}$	$\frac{1}{24}$	0
$h^4f_jf_kf_l^kf_i^l$	$\frac{1}{24}$	$dtrm$	$\frac{1}{24}$	0
$h^5f_jf_kf_lf_mf_i^{jklm}$	$\frac{1}{120}$	$\frac{1}{24}(bm^4+cn^4+dp^4)$	$\frac{5}{576}$	$+\frac{1}{2880}$
$h^5f_jf_kf_lf_m^lf_i^{km}$	$\frac{1}{20}$	$\frac{1}{2}\{crmn^2+d(sm+tn)p^2\}$	$\frac{5}{96}$	$+\frac{1}{480}$
$h^5f_jf_kf_lf_m^kf_i^{lm}$	$\frac{1}{30}$	$\frac{1}{2}\{crm^2n+d(sm^2+tn^2)p\}$	$\frac{1}{32}$	$-\frac{1}{480}$
$h^5f_jf_kf_l^kf_m^lf_i^m$	$\frac{1}{30}$	$dtrmp$	$\frac{1}{24}$	$+\frac{1}{120}$
$h^5f_jf_kf_lf_m^{jkl}f_i^m$	$\frac{1}{120}$	$\frac{1}{6}\{crm^3+d(sm^3+tn^3)\}$	$\frac{1}{144}$	$-\frac{1}{720}$
$h^5f_jf_kf_l^kf_m^lf_i^m$	$\frac{1}{40}$	$dtrmn$	$\frac{1}{48}$	$-\frac{1}{240}$
$h^5f_jf_kf_l^mf_m^jf_i^kf_i^l$	$\frac{1}{120}$	$\frac{1}{2}dtrm^2$	$\frac{1}{96}$	$+\frac{1}{480}$
$h^5f_jf_k^jf_l^kf_m^lf_i^m$	$\frac{1}{120}$	0	0	$-\frac{1}{120}$
$h^5f_jf_kf_l^jf_m^kf_i^{lm}$	$\frac{1}{40}$	$\frac{1}{2}\{cr^2m^2+d(sm+tn)^2\}$	$\frac{1}{24}+\frac{1}{48}\sqrt{\frac{1}{2}}$	$\frac{1}{60}+\frac{1}{48}\sqrt{\frac{1}{2}}$

$$\left.\begin{aligned} a+b+c+d &= 1,\\ bm+cn+dp &= \tfrac{1}{2},\\ bm^2+cn^2+dp^2 &= \tfrac{1}{3},\\ cmr+d(nt+ms) &= \tfrac{1}{6},\\ bm^3+cn^3+dp^3 &= \tfrac{1}{4},\\ cmnr+d(nt+ms)p &= \tfrac{1}{8},\\ cm^2r+d(n^2t+m^2s) &= \tfrac{1}{12},\\ dmrt &= \tfrac{1}{24}, \end{aligned}\right\} \tag{13}$$

which imply

$$p = 1. \tag{14}$$

We have here two degrees of freedom. It is possible to express each quantity rationally in terms of m and n, but the expressions are clumsy and difficult to use in any further analysis. However, Kutta suggested five special cases in which, whilst one degree of freedom is retained, the quantities can be expressed in particularly simple forms. These are listed in Table 2. The Runge-Kutta rule, the simplest particular solution, may be obtained by putting $m = \frac{1}{2}$ in case (i), or $t = 1$ in case (v).

The main disadvantage of this type of process is that four substitutions have to be made into the given equations (10) for each step of the integration, and in this respect it compares unfavourably with other types of similar accuracy. This is part of the price that has to be paid for abandoning preceding function values, and if the equations (10) are complicated this may be a serious drawback. However, systems consisting of many equations are, in practice, usually simple in form, or may be made so by the introduction of additional variables, using techniques similar to those employed on the Differential Analyser.

Table 2

Quantity	Case (i)	Case (ii)	Case (iii)	Case (iv)	Case (v)
m	m	m	$\frac{1}{2}$	1	$\frac{1}{2}$
n	$1-m$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
p	1	1	1	1	1
r	$\frac{1-m}{2m}$	$\frac{1}{8m}$	r	$\frac{1}{8}$	$\frac{1}{2t}$
s	$\frac{(1-m)(2m-1)}{2m[6m(1-m)-1]}$	$-\frac{1}{2m}$	$\frac{3}{2}$	$-\frac{t}{4}$	$1-t$
t	$\frac{m}{6m(1-m)-1}$	2	$\frac{1}{2r}$	t	t
a	$\frac{6m(1-m)-1}{12m(1-m)}$	$\frac{1}{6}$	$\frac{1}{6}-\frac{1}{12r}$	$\frac{1}{6}$	$\frac{1}{6}$
b	$\frac{1}{12m(1-m)}$	0	$\frac{2}{3}$	$\frac{1}{6}-\frac{1}{3t}$	$\frac{2-t}{3}$
c	$\frac{1}{12m(1-m)}$	$\frac{2}{3}$	$\frac{1}{12r}$	$\frac{2}{3}$	$\frac{t}{3}$
d	$\frac{6m(1-m)-1}{12m(1-m)}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{3t}$	$\frac{1}{6}$

3. *Choice of process.* We have now to consider the number of storage registers per equation required by the process as given by (12). At the beginning of the step we need $n+1$ registers to hold the quantities $Y_0, \ldots, Y_n$, i.e. one register per equation. For the first stage we need one more register per equation, to receive the quantities k_{i0}. For the second stage we need $Y_i + mk_{i0}$, and a register to receive k_{i1}. We must also store in some form $Y_i + [n-r]\,k_{i0}$ which will be needed at the next stage, $Y_i + [p-s-t]\,k_{i0}$ which will be needed at the last stage, and $Y_i + ak_{i0}$ which will be needed at the end of the step. However, these five quantities are linearly dependent and can certainly be represented by three, and so three registers will suffice for the second stage.

It is at the third stage that the maximum number of registers is required. We have to store in some form

$$Y_i + [n-r]\,k_{i0} + rk_{i1}, \quad Y_i + [p-s-t]\,k_{i0} + sk_{i1}, \quad Y_i + ak_{i0} + bk_{i1} \quad \text{and} \quad k_{i2},$$

and in general four registers are necessary to each equation. At the last stage three are certainly sufficient, because we do not have to consider forming a value of y_i for a following stage. Thus if we can reduce the number of registers required in the third stage to three per equation, we need never exceed this number in the whole process.

Clearly, three registers will suffice for the third stage if the quantities to be stored are linearly dependent, i.e. if

$$\begin{vmatrix} 1 & n-r & r \\ 1 & p-s-t & s \\ 1 & a & b \end{vmatrix} = 0. \tag{15}$$

This condition is not incompatible with (13), and the combined equations have a single infinity of solutions. To find the simplest particular solutions we can apply (15) to each of the cases listed in Table 2. The results are as follows.

Case (i). $3m^3 - 9m^2 + 6m - 1 = 0$, whence

$$m = 1 + \sqrt{\tfrac{4}{3}}\cos(120N + 10)^\circ = 0{\cdot}258,\ 0{\cdot}605 \text{ or } 2{\cdot}137.$$

None of these cases is very attractive as the basis of a practical process.

Case (ii). There is no solution.

Case (iii). $4r^2 + 4r + 5 = 0$. There is no real solution.

Case (iv). $2t^2 - 6t + 5 = 0$. There is no real solution.

Case (v). $2t^2 - 4t + 1 = 0$, whence $t = 1 \pm \sqrt{\tfrac{1}{2}}$.

The corresponding values for the other constants are

$$\left.\begin{array}{llll} m = \tfrac{1}{2}, & r = 1 \mp \sqrt{\tfrac{1}{2}}, & a = \tfrac{1}{6}, & \\ n = \tfrac{1}{2}, & s = \mp \sqrt{\tfrac{1}{2}}, & b = \tfrac{1}{3}(1 \mp \sqrt{\tfrac{1}{2}}), & \\ p = 1, & t = 1 \pm \sqrt{\tfrac{1}{2}}, & c = \tfrac{1}{3}(1 \pm \sqrt{\tfrac{1}{2}}), & \\ & & d = \tfrac{1}{6}. & \end{array}\right\} \tag{16}$$

These two solutions are probably the simplest that exist. They possess, in common with the Runge-Kutta case, the advantage that the independent variable assumes only values corresponding to the beginning and end of each step, and the midpoint of each

step. This is valuable in cases where the equations involve a function of x which is defined by a table at equal intervals.

To decide which form of (16) to use, we appeal to the fifth-order terms in the expansion of y. The last two columns of Table 1 show that only the last of these terms is affected by our choice, and clearly indicate the adoption of the upper sign for the square root throughout (16). The rest of the paper will deal solely with this case.

Substituting the values given by (16) into the equations (12), and introducing new symbols for the intermediate values of y, we have

$$\left.\begin{aligned} k_{i0} &= hf_i(y_{00}, y_{10}, y_{20}, \ldots), \\ k_{i1} &= hf_i(y_{01}, y_{11}, y_{21}, \ldots), \\ k_{i2} &= hf_i(y_{02}, y_{12}, y_{22}, \ldots), \\ k_{i3} &= hf_i(y_{03}, y_{13}, y_{23}, \ldots), \end{aligned}\right\} \tag{17}$$

where

$$\left.\begin{aligned} y_{i0} &= Y_i, \\ y_{i1} &= Y_i + \tfrac{1}{2}k_{i0}, \\ y_{i2} &= Y_i + [-\tfrac{1}{2} + \sqrt{\tfrac{1}{2}}]\,k_{i0} + [1 - \sqrt{\tfrac{1}{2}}]\,k_{i1}, \\ y_{i3} &= Y_i + [-\sqrt{\tfrac{1}{2}}]\,k_{i1} + [1 + \sqrt{\tfrac{1}{2}}]\,k_{i2}, \end{aligned}\right\} \tag{18}$$

and the adopted value of y_i at the end of the step is

$$y_{i4} = Y_i + \tfrac{1}{6}k_{i0} + \tfrac{1}{3}[1 - \sqrt{\tfrac{1}{2}}]\,k_{i1} + \tfrac{1}{3}[1 + \sqrt{\tfrac{1}{2}}]\,k_{i2} + \tfrac{1}{6}k_{i3}. \tag{18a}$$

When calculating k_{ip} $(i = 0, \ldots, n)$ for a particular p, we require a register to hold each y_{ip}, and a register to receive each k_{ip}. For the quantities stored in the third set of registers, any linear combination of Y_i and the preceding values of k_i, such as will enable us to derive the quantities required in succeeding stages, will meet the case. There is, however, as will appear later, some advantage in choosing a quantity which is purely a combination of the preceding values of k_i and does not contain Y_i; such a quantity will therefore be of the order of h. This determines it apart from an arbitrary factor, and we shall adopt the following:

$$\left.\begin{aligned} q_{i1} &= k_{i0}, \\ q_{i2} &= [-2 + 3\sqrt{\tfrac{1}{2}}]\,k_{i0} + [2 - \sqrt{2}]\,k_{i1}, \\ q_{i3} &= [-\tfrac{1}{2}]\,k_{i0} + [-1 - \sqrt{2}]\,k_{i1} + [2 + \sqrt{2}]\,k_{i2}. \end{aligned}\right\} \tag{19}$$

Introducing the quantities defined by (18) and (19) into the process as given by (17), we obtain

$$\left.\begin{aligned} & & k_{i0} &= hf_i(y_{00}, y_{10}, \ldots), & & \\ y_{i1} &= y_{i0} + \tfrac{1}{2}k_{i0}, & & & q_{i1} &= k_{i0}, \\ & & k_{i1} &= hf_i(y_{01}, y_{11}, \ldots), & & \\ y_{i2} &= y_{i1} + [1 - \sqrt{\tfrac{1}{2}}](k_{i1} - q_{i1}), & & & q_{i2} &= [2 - \sqrt{2}]\,k_{i1} + [-2 + 3\sqrt{\tfrac{1}{2}}]\,q_{i1}, \\ & & k_{i2} &= hf_i(y_{02}, y_{12}, \ldots), & & \\ y_{i3} &= y_{i2} + [1 + \sqrt{\tfrac{1}{2}}](k_{i2} - q_{i2}), & & & q_{i3} &= [2 + \sqrt{2}]\,k_{i2} + [-2 - 3\sqrt{\tfrac{1}{2}}]\,q_{i2}, \\ & & k_{i3} &= hf_i(y_{03}, y_{13}, \ldots), & & \\ y_{i4} &= y_{i3} + \tfrac{1}{6}k_{i3} - \tfrac{1}{3}q_{i3}. & & & & \end{aligned}\right\} \tag{20}$$

The sequence of events during one step is as follows:

(i) Calculate k_{i0} $(i = 0, \ldots, n)$.
(ii) Calculate y_{i1} and q_{i1} $(i = 0, \ldots, n)$. The most convenient order in which to do this is $y_{01}, q_{01}; y_{11}, q_{11}; y_{21}, q_{21}; \ldots$.
(iii) Calculate k_{i1} $(i = 0, \ldots, n)$.
(iv) Calculate y_{i2} and q_{i2} $(i = 0, \ldots, n)$.

.....

(viii) Calculate y_{14} $(i = 0, \ldots, n)$.

As each quantity is calculated it is stored in the register formerly holding the corresponding quantity of the previous stage, which is no longer required.

It would be quite possible to use the process in this form, but one or two refinements can be made to give greater accuracy whilst using the same storage space, with little increase in the complexity of the process. The rest of the paper is devoted to the development and study of the more accurate form.

4. *Obtaining maximum accuracy.* The process as it stands suffers from the disadvantage that many rounding-off errors are accumulated during each step. If the error committed in the determination of each of the eleven quantities in (20) is represented by the operator e, then the total error in y_i accumulated during one step is

$$\begin{aligned} e(\delta y_i) = {} & \tfrac{1}{6}e(k_{i0}) + \tfrac{1}{3}[1 - \sqrt{\tfrac{1}{2}}]\, e(k_{i1}) + \tfrac{1}{3}[1 + \sqrt{\tfrac{1}{2}}]\, e(k_{i2}) + \tfrac{1}{6}e(k_{i3}) \\ & + e(y_{i1}) + e(y_{i2}) + e(y_{i3}) + e(y_{i4}) \\ & - \tfrac{1}{3}e(q_{i1}) - \tfrac{1}{3}e(q_{i2}) - \tfrac{1}{3}e(q_{i3}). \end{aligned} \tag{21}$$

Now the quantities k are small, being of the order of h. It is possible therefore to store them to a higher degree of absolute accuracy than can be achieved in y, by introducing a scale factor g, i.e. by storing $g^{-1}k$ instead of k. If we assume that the same number of digits is available for k as for y, then g can be made comparable with h without causing $g^{-1}k$ to exceed the capacity of the register. Moreover, the extra digits thus stored are significant provided that the f_i^j are of the order of unity or less, for if this condition holds, then f_i can be obtained to the same order of accuracy as y, and k will have an error of the order of h times that of y. The same factor may be applied to q, because

$$q_{i1} = k_{i0}, \quad q_{i2} \sim (\sqrt{\tfrac{1}{2}})\, k, \quad q_{i3} \sim \tfrac{1}{2}k, \tag{22}$$

and hence if $g^{-1}k$ is within capacity, so is $g^{-1}q$. The developments now to be described depend on the fact that k and q are stored in this way.

The rounding-off errors of k and q may thus be made negligible compared with those of y, which are then the only large terms remaining on the right-hand side of (21). However, we are still at liberty to adjust q if we please, and the possibility now arises of introducing 'deliberate errors' in the estimations of q, to compensate almost exactly for the rounding-off errors which we are forced to make in y. This is possible because we can choose q from an array of values which has a much closer spacing than the available values for representing y. In fact, if we make the errors in q satisfy as nearly as possible the following:

$$e(q_{i1}) = 3e(y_{i1}), \quad e(q_{i2}) = 3e(y_{i2}), \quad e(q_{i3}) = 3e(y_{i3}), \tag{23}$$

then the resultant $e(\delta y_i)$ as given by (21) contains only quantities of the same order as the rounding-off errors of k and q, with the exception of the single term $e(y_{i4})$. Before considering how further reductions of error may be made, let us see how the process appears with these modifications.

We first define the auxiliary quantities

$$\left.\begin{aligned} r_{i1} &= y_{i1}-y_{i0} = \tfrac{1}{2}k_{i0}+e(r_{i1}),\\ r_{i2} &= y_{i2}-y_{i1} = [1-\sqrt{\tfrac{1}{2}}]\,(k_{i1}-q_{i1})+e(r_{i2}),\\ r_{i3} &= y_{i3}-y_{i2} = [1+\sqrt{\tfrac{1}{2}}]\,(k_{i2}-q_{i2})+e(r_{i3}). \end{aligned}\right\} \tag{24}$$

The last term in each case represents an error which is deliberately introduced in order to round-off r to the same digital accuracy as y. The new value of y is now found by adding r to the previous value, no further rounding-off being necessary; in fact, $e(r_{ip}) = e(y_{ip})$. If now we arrange that the corresponding q contains a term $3r$, we shall have achieved our object. The values of q are now defined thus:

$$\left.\begin{aligned} q_{i1} &= 3r_{i1}-\tfrac{1}{2}k_{i0},\\ q_{i2} &= q_{i1}+3r_{i2}-[1-\sqrt{\tfrac{1}{2}}]\,k_{i1},\\ q_{i3} &= q_{i2}+3r_{i3}-[1+\sqrt{\tfrac{1}{2}}]\,k_{i2}. \end{aligned}\right\} \tag{25}$$

Note that the quantities r are only used temporarily during the formation of each y and q; we do not need an extra register for each variable.

We now return to consider the remaining large term in $e(\delta y_i)$, namely, $e(y_{i4})$. With our present scheme, $e(y_{i4})$ can be neither eliminated nor compensated for because we have no quantity q_{i4} which can be adjusted accordingly. We have, in fact, achieved a high degree of accuracy within each step, but we are throwing away some of this accuracy at the end of the step by rounding-off y_{i4} without keeping any record of the rounding-off error. There is, however, no objection to introducing a quantity q_{i4}, solely for the purpose of retaining the high accuracy of the calculation from one step to the next. This 'bridging' value of q will be much smaller than the values encountered within the step, and in fact it will be equal, as nearly as possible, to three times the rounding-off error of y_{i4}. It becomes q_{i0} for the following step.

We now have:

$$\left.\begin{aligned} k_{i0} &= hf_i(y_{00}, y_{10}, \ldots), & r_{i1} &= \tfrac{1}{2}k_{i0}-wq_{i0}+e(r_{i1}),\\ y_{i1} &= y_{i0}+r_{i1}, & q_{i1} &= q_{i0}+3r_{i1}-\tfrac{1}{2}k_{i0},\\ k_{i1} &= hf_i(y_{01}, y_{11}, \ldots), & r_{i2} &= [1-\sqrt{\tfrac{1}{2}}]\,(k_{i1}-q_{i1})+e(r_{i2}),\\ y_{i2} &= y_{i1}+r_{i2}, & q_{i2} &= q_{i1}+3r_{i2}-[1-\sqrt{\tfrac{1}{2}}]\,k_{i1},\\ k_{i2} &= hf_i(y_{02}, y_{12}, \ldots), & r_{i3} &= [1+\sqrt{\tfrac{1}{2}}]\,(k_{i2}-q_{i2})+e(r_{i3}),\\ y_{i3} &= y_{i2}+r_{i3}, & q_{i3} &= q_{i2}+3r_{i3}-[1+\sqrt{\tfrac{1}{2}}]\,k_{i2},\\ k_{i3} &= hf_i(y_{03}, y_{13}, \ldots), & r_{i4} &= \tfrac{1}{6}(k_{i3}-2q_{i3})+e(r_{i4}),\\ y_{i4} &= y_{i3}+r_{i4}, & q_{i4} &= q_{i3}+3r_{i4}-\tfrac{1}{2}k_{i3}. \end{aligned}\right\} \tag{26}$$

The coefficient w in the expression for r_{i1} is not very critical. It appears at first sight that we should take $w = \frac{1}{3}$, for this would give us the best possible estimate for y_{i1}, but it will be shown later that in fact the overall optimum value is $w = 1$. However,

since q_{i0} is of the order of the last digit of y, and errors in r are compensated for over the step, we introduce no appreciable error by changing w, if this is convenient. The error thus introduced is considered in the next section.

This modification has not destroyed the facility of automatic starting possessed by the process. The bridging q serves merely to ensure that rounding-off errors in y do not build up appreciably over a large range of integration. At the beginning of a range we can with little loss of accuracy set $q_{i0} = 0$; in fact, in many cases, the initial value of y_i will be given exactly with no rounding-off error. The facility of being able to change h between any two steps is still available, provided we do not wish to change g. If g is changed we must either change every $g^{-1}q_{i0}$ correspondingly, or suffer a small persistent error of the order of the rounding-off error in y times the relative change in g.

The quantities q now play a double role; they serve to give the process the analytical form of (20), and they also effectively carry a few extra digits of y. The process is here in its final form. Apart from the economy of registers, two features make the process particularly suitable for use in an automatic machine: first, since each value of k, y and q may be 'written over' its predecessor, wasteful shifting operations are avoided, and secondly, the four stages composing one step are very similar in form and can be carried out by substantially the same sequence of instructions.

5. *Analysis of errors.* We now have to make a more thorough study of the errors occurring in the final form of the process. The quantity with which we are primarily concerned now is the value of $y_{i4} - \frac{1}{3}q_{i4}$ at the end of each step, for this represents the nearest estimate we possess to the analytical solution. The quantity we denote by y_i will have rounding-off errors, but these will not accumulate; they will nearly cancel out over a short range, and the cumulative error over a large range will be the sum of the errors in $y_{i4} - \frac{1}{3}q_{i4}$.

These errors are of two kinds: those due to rounding-off, and 'truncation' errors caused by the fact that the process yields only fourth-order accuracy. The latter have been listed in Table 1, and little more need be said about them here. It must be remembered that each of the fifth-degree terms in that table is the sum of $(n+1)^4$ products of the functions f and their derivatives. In practice, however, where n is large, such as in the solution of a partial differential equation as a set of ordinary equations, the majority of these products will vanish. In most cases the last term but one will probably be dominant, since it contains only first-order derivatives.

Coming now to rounding-off errors, let us consider first the errors introduced by using rounded-off forms of the coefficients $\sqrt{\frac{1}{2}}$ and $\frac{1}{6}$. If $\sqrt{\frac{1}{2}}$ is replaced by $\sqrt{\frac{1}{2}} + e_1$, investigation shows that the resultant error in $y_{i4} - \frac{1}{3}q_{i4}$ is $\frac{1}{3}(k_{i2} - k_{i1})\, e_1$. Expressed as a power series in h, the largest term is

$$\tfrac{1}{6}[1 - \sqrt{\tfrac{1}{2}}]\, h^3 f_j f_k^j f_i^k e_1.$$

Thus if $\sqrt{\frac{1}{2}}$ is stored to the same accuracy as y, the resultant error per step is of the order of h^3u, where u is the value of one unit in the last digit of y; over a finite range the error is $O(h^2u)$ and is quite negligible.

If $\frac{1}{6} + e_2$ is used in place of $\frac{1}{6}$ in forming r_{i4}, no error is produced in $y_{i4} - \frac{1}{3}q_{i4}$, i.e. no cumulative error is caused directly. However, indirect errors arise from using in-

correct values of y_{i4} in the calculation of k_{j4}. The effect of the errors in y_i on the values of k_j will be considered shortly, but for the present we need only note that the effect of e_2 will be entirely negligible. Owing to rounding-off it is meaningless to attempt to keep the error in y_{i4} much below about $\frac{1}{2}u$. The error in y_{i4} due to e_2 is actually

$$(k_{i0}+[2+2\sqrt{2}]\,k_{i1}+[-4-2\sqrt{2}]\,k_{i2}+k_{i3})\,e_2,$$

which is of the order of h^3e_2.

We now proceed to estimate the effect on $y_{i4}-\frac{1}{3}q_{i4}$ of rounding-off errors in the other quantities. The accumulated error is due entirely to the errors in k_i and q_i, because the terms in $e(r_i)$ cancel out. However, errors in k_i are caused partly by using erroneous values of y_j, and so it is convenient to begin by studying the behaviour of y itself. If we assume that $y_{i0}-\frac{1}{3}q_{i0}$ represents the correct value of the function at the beginning of the step, the differences between the stored values y_i and the correct solution are due almost entirely to the terms $e(r_i)$, which are larger than the errors in k_i and q_i by a factor of the order of h^{-1}. Now we have seen that each $e(r_i)$ has no direct effect on quantities in following steps. It can and does, however, affect following values of r_i, q_i and y_i up to the end of the present step. Thus $e(r_{i1})$, for example, affects not only y_{i1} but y_{i2} and y_{i3} as well. The errors in y within one step are therefore not independent, and in fact we shall find that they tend to cancel one another out. Table 3 shows the effect on each quantity within one step of the errors $e(r_{i0})$, $e(r_{i1})$, $e(r_{i2})$ and $e(r_{i3})$, where r_{i0} is the quantity denoted by r_{i4} in the preceding step.

If we represent total errors by E, the error in any k_i due to errors in y_j is

$$h\sum_j f_i^j E(y_j).$$

The error in $y_{i4}-\frac{1}{3}q_{i4}$ due to errors in k_i is, by (18a),

$$\tfrac{1}{6}E(k_{i0})+\tfrac{1}{3}[1-\sqrt{\tfrac{1}{2}}]\,E(k_{i1})+\tfrac{1}{3}[1+\sqrt{\tfrac{1}{2}}]\,E(k_{i2})+\tfrac{1}{6}E(k_{i3}), \tag{27}$$

and the contribution to this due to errors in y_j is therefore

$$h\sum_j f_i^j\{\tfrac{1}{6}E(y_{j0})+\tfrac{1}{3}[1-\sqrt{\tfrac{1}{2}}]\,E(y_{j1})+\tfrac{1}{3}[1+\sqrt{\tfrac{1}{2}}]\,E(y_{j2})+\tfrac{1}{6}E(y_{j3})\},$$

Table 3

Quantity	Coefficient of errors			
	$e(r_{i0})$	$e(r_{i1})$	$e(r_{i2})$	$e(r_{i3})$
r_{i0}	1	0	0	0
q_{i0}	3	0	0	0
y_{i0}	1	0	0	0
r_{i1}	$-3w$	1	0	0
q_{i1}	$3(1-3w)$	3	0	0
y_{i1}	$1-3w$	1	0	0
r_{i2}	$-3(1-\sqrt{\frac{1}{2}})\,(1-3w)$	$-3(1-\sqrt{\frac{1}{2}})$	1	0
q_{i2}	$3(-2+3\sqrt{\frac{1}{2}})\,(1-3w)$	$3(-2+3\sqrt{\frac{1}{2}})$	3	0
y_{i2}	$(-2+3\sqrt{\frac{1}{2}})\,(1-3w)$	$(-2+3\sqrt{\frac{1}{2}})$	1	0
r_{i3}	$3(\frac{1}{2}-\sqrt{\frac{1}{2}})\,(1-3w)$	$3(\frac{1}{2}-\sqrt{\frac{1}{2}})$	$-3(1+\sqrt{\frac{1}{2}})$	1
q_{i3}	$-\frac{3}{2}(1-3w)$	$-\frac{3}{2}$	$-3(2+3\sqrt{\frac{1}{2}})$	3
y_{i3}	$-\frac{1}{2}(1-3w)$	$-\frac{1}{2}$	$-(2+3\sqrt{\frac{1}{2}})$	1
r_{i4}	$\frac{1}{2}(1-3w)$	$\frac{1}{2}$	$(2+3\sqrt{\frac{1}{2}})$	-1
q_{i4}	0	0	0	0
y_{i4}	0	0	0	0

neglecting changes in f_i^j during the step. Substituting the expressions for $E(y_j)$ from Table 3 we obtain

$$h\sum_j f_i^j\{\tfrac{1}{4}(1-w)\,e(r_{j0})+\tfrac{1}{12}e(r_{j1})-\tfrac{1}{6}\sqrt{\tfrac{1}{2}}\,e(r_{j2})+\tfrac{1}{6}e(r_{j3})\}, \tag{28}$$

which is somewhat smaller than would be obtained if the errors in y_j were independent. Assuming that the values of $e(r_j)$ are independent and randomly distributed between $-\frac{1}{2}u$ and $+\frac{1}{2}u$, we find that the standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ from this source is equal to

$$\tfrac{1}{24}[(3[1-w]^2+\tfrac{7}{3})\sum_j (f_i^j)^2]^{\frac{1}{2}}\,hu. \tag{29}$$

This expression leads to the optimum value of $w = 1$ quoted in the last section.

It remains to estimate the errors due to rounding-off k_i and q_i. The behaviour of k may be abnormal in some cases, depending on the form of the given functions f, but we shall assume that the rounding-off errors in k_i are randomly distributed between $-\frac{1}{2}gu$ and $+\frac{1}{2}gu$. By (27) they cause a standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ equal to $\frac{1}{6}\sqrt{\frac{7}{6}}\,gu$.

The rounding-off errors in q_i are simply additive over the step. Referring to (26), we see that they are due entirely to the last term in each expression for q_i, since, assuming that we have chosen g^{-1} to be an integer, the first two terms are both exact to the number of digits to which $g^{-1}q_i$ is stored. In q_{i2} and q_{i3} the coefficient of k_i in the last term is not a simple ratio, and therefore the rounding-off error will be random between $-\frac{1}{2}gu$ and $+\frac{1}{2}gu$. In q_{i1} and q_{i4}, however, the coefficient is in each case $-\frac{1}{2}$, and we cannot round-off without introducing a bias of $+\frac{1}{4}gu$ or $-\frac{1}{4}gu$. It is important that the total bias should be removed by rounding-off q_{i1} and q_{i4} in opposite directions, one up and one down. The standard deviation in each case is $\frac{1}{4}gu$. The resultant standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ from the rounding-off of q_i is thus $\frac{1}{6}\sqrt{\frac{7}{6}}gu$.

The combined standard deviation in $y_i-\frac{1}{3}q_i$ over one step from all rounding-off errors is

$$\tfrac{1}{6}[\tfrac{7}{3}g^2+\tfrac{1}{16}(3[1-w]^2+\tfrac{7}{3})\,h^2\sum_j (f_i^j)^2]^{\frac{1}{2}}\,u. \tag{30}$$

In most cases $\Sigma(f_i^j)^2$ will be sufficiently small to make the second term in the root unimportant, and the value of w will not appreciably affect the accuracy of the process. If $\Sigma(f_i^j)^2$ is exceptionally large the effect of taking, say, $w = 0$ instead of $w = 1$ may just be noticeable.

If $w = 1$, (30) becomes

$$\tfrac{1}{6}[\tfrac{7}{3}\{g^2+\tfrac{1}{16}h^2\sum_j (f_i^j)^2\}]^{\frac{1}{2}}\,u, \tag{31}$$

which is the same standard deviation as would be caused by a single rounding-off to the nearest multiple of

$$v = \tfrac{1}{3}[7\{g^2+\tfrac{1}{16}h^2\sum_j (f_i^j)^2\}]^{\frac{1}{2}}\,u \simeq gu. \tag{32}$$

The overall accuracy of the process, if truncation errors are negligible, is therefore approximately the same as would be obtained if y were rounded-off at each step to the nearest multiple of gu, so that we are effectively keeping a number of extra digits of y corresponding to the value of g. It is important to note that q_{i1} and q_{i4} must be oppositely rounded-off to eliminate bias.

6. *Example.* As a simple example, the single equation $y' = y$ will be integrated from $x = 0$ to $x = 1$ in steps of $h = 0{\cdot}1$. We shall suppose that we are using a decimal machine,

with registers of six-figure capacity, and that the decimal point is at the left-hand end of the register. The starting value will be $y(0) = 0{\cdot}1$. We can take $g = h = 0{\cdot}1$, for this will make $g^{-1}k = g^{-1}hy' = y$, so that $g^{-1}k$ will be within capacity. We shall take the value 0·7071 for $\sqrt{\frac{1}{2}}$; the error is about 7×10^{-6}, which is just small enough to have negligible effect. For $\frac{1}{6}$, the value 0·1667 is sufficiently accurate. We shall use the case $w = 1$.

Table 4

x	Stage	r	$y = g^{-1}k$	$g^{-1}q$
0·0	0		100 000	0
	1	5 000	105 000	100 000
	2	146	105 146	73 626
	3	5 381	110 527	55 561
0·1	4	− 10	110 517	− 3
	1	5 526	116 043	110 519
	2	162	116 205	81 390
	3	5 943	122 148	61 306
0·2	4	− 8	122 140	− 8
	1	6 108	128 248	122 162
	2	178	128 426	89 938
	3	6 570	134 996	67 802
0·3	4	− 10	134 986	+ 4
	1	6 749	141 735	134 981
	2	198	141 933	99 407
	3	7 260	149 193	74 913
0·4	4	− 11	149 182	− 14
	1	7 461	156 643	149 225
	2	217	156 860	109 854
	3	8 024	164 884	82 798
0·5	4	− 12	164 872	− 4
	1	8 244	173 116	164 880
	2	241	173 357	121 404
	3	8 869	182 226	91 536
0·6	4	− 14	182 212	+ 3
	1	9 110	191 322	182 197
	2	267	191 589	134 169
	3	9 802	201 391	101 167
0·7	4	− 16	201 375	− 9
	1	10 070	211 445	201 404
	2	294	211 739	148 292
	3	10 831	222 570	111 762
0·8	4	− 16	222 554	− 3
	1	11 128	233 682	222 560
	2	326	234 008	163 895
	3	11 969	245 977	123 490
0·9	4	− 17	245 960	− 9
	1	12 299	258 259	245 981
	2	360	258 619	181 137
	3	13 227	271 846	136 459
1·0	4	− 18	271 828	− 4

The truncation error will consist entirely of the last term but one in Table 1, which gives

$$-\tfrac{1}{120}h^5y = -8{\cdot}3\times10^{-8}y \quad \text{per step,}$$

i.e.

$$-8{\cdot}3\times10^{-8}\Sigma y = -14\times10^{-8} \quad \text{over the range.}$$

In this particular example there is no rounding-off error introduced by k. Under this condition, the theory of the last section predicts a standard deviation of

$$\tfrac{1}{24}\sqrt{(21)}gu = 2\times10^{-8} \quad \text{over one step,}$$

i.e.

$$\sqrt{(10)}\times2\times10^{-8} = 6\times10^{-8} \quad \text{over the range.}$$

To remove bias, q_1 will be rounded up and q_4 down. Since $g^{-1}k = y$, the common value need only be recorded once.

The results of the calculation are shown in Table 4. It will be seen that the value of $y - \frac{1}{3}q$ at the end of the range is 0·27182813, which has an error of -5×10^{-8}.

7. *Application.* A routine has been prepared by the author for applying the process on the EDSAC (Electronic Delay Storage Automatic Calculator) at the Mathematical Laboratory. The routine handles any number of equations within the storage capacity of the machine; it consists of sixty-seven instructions (single address), and has to be supplemented by a routine which evaluates every k as required by the given system of equations. The time required to perform each step is 0·21 sec. per variable, plus four times the time required to evaluate the k's.

Acknowledgements. The author wishes to thank Mr D. J. Wheeler for many helpful suggestions, Dr M. V. Wilkes for advice in preparing the manuscript, and the Department of Scientific and Industrial Research for a Maintenance Allowance.

REFERENCE

(1) KUTTA, W. *Z. Math. Phys.* 46 (1901), 435.

UNIVERSITY MATHEMATICAL LABORATORY
CAMBRIDGE