AN OPERATIONAL METHOD FOR THE STUDY OF INTEGRATION PROCESSES

by

R.H. Merson
Royal Aircraft Establishment, Farnborough, England

1. INTRODUCTION

The increasing complexity and diversity of physical systems that are now being studied, coupled with the advent of more and better high-speed digital computers to aid the applied mathematician, has necessitated a reconsideration of many of the basic ideas concerning methods of numerical analysis. Essentially new requirements have to be met and methods of analysis and computation suitable for hand - or desk - computing are, in many cases, not at all suitable for use with a digital computer, generally because of the moronic nature of the latter. In the field of ordinary differential equations the Central Difference and other similar methods have given way to the Runge-Kutta process, but this and other fourth order Kutta processes have disadvantages which it is not proposed to discuss here. It is a fact, however, that there remains a tremendous amount of basic research to be done in studying and classifying the characteristics of the many hundreds of possible step-by-step integration processes. Among the factors which have to be considered in any such classification which is to be comprehensive are the order of a process and the number of starting points required, its stability, elementary truncation error, susceptibility to round-off errors, behaviour in passing through discontinuities, the ease with which running error estimates can be made and its relative speed of operation when programmed for a digital computer.

This paper is concerned with the evaluation of elementary truncation errors and in particular with the regularisation of an operational calculus for handling the large sets of multiple differential operators which arise in the development of new types of integration processes. The usual notation for such operators has been found to be far too cumbersome when handling fourth-or higher-order operators, and an abbreviated notation has been introduced. The development of this new notation has led to the discovery of a one-one correspondence between certain multiple differential operators and the zero-order linear graphs or "trees" occurring in topology (Reference 1,2). This correspondence provides a means of picturing or visualising the operators and has been found useful in several ways, for example, in producing a multiplication table for basic operators.

2. MULTIPLE DIFFERENTIAL OPERATORS

Consider a set of n first order ordinary differential equations

$$\frac{dy_i}{dt} = f_i\,(y_1, y_2, \ldots, y_n), \quad i = 1,2, \ldots,n, \tag{1}$$

where, if required, the independent variable t may be taken to be one of the y_i.

It will be assumed that the functions f_i are continuous and differentiable with respect to all the y_i as many times as necessary.

These equations are to be integrated by some linear step-by-step integration process.

In order to introduce the differential operators that arise the simple process of linear extrapolation will be considered, namely

$$y_{i,s+1} = y_{i,s} + hf_{i,s}$$

where $y_{i,s}$ is the computed value of y_i at the end of the s-th step,

$$f_{i,s} = f_i\,(y_{1,s}\,,\, y_{2,s}\,,\, \dots,\, y_{n,s}\,),$$

and $h = t_s - t_{s-1}$ is the interval of integration, which is assumed to be constant.

It follows that

$$f_{i,s+1} = f_i\,(y_{1,s+1}\,,\, \dots,\, y_{n,s+1})$$

$$= f_i\,(Y_{1,s} + hf_{1,s}\,,\, \dots,\, y_{n,s} + hf_{n,s}\,),$$

and expanding in a Taylor series,

$$f_{i,s+1} = f_{i,s} + h\sum_j (f_j\, f_i^{\,j})_s + \frac{1}{2!}\, h^2 \sum_{j,k} (f_j\, f_k\, f_i^{\,jk})_s$$

$$+ \frac{1}{3!}\, h^3 \sum_{j,k,\ell} (f_j\, f_k\, f_\ell\, f_i^{\,jk\ell})_s + O(h^4),$$

where $\quad f_i^{\,j} = \dfrac{\partial f_i}{\partial y_j}\,,\quad f_i^{\,jk} = \dfrac{\partial^2 f_i}{\partial y_j\, \partial y_k}\,,$ etc.

As a first step in simplifying the above somewhat cumbersome notation, let us define the following sequence of operators:

$$\sum_j f_j\,(\ \ )^j = D$$

$$\sum_{j,k} f_j f_k\,(\ \ )^{jk} = D^{(2)}$$

$$\sum_{j,k,\ell} f_j f_k f_\ell\,(\ \ )^{jk\ell} = D^{(3)}$$

Then it is seen that the above expression for $f_{i,s+1}$ can be written symbolically as

$$f_{i,s+1} = \Big[\exp(hD) f_i\Big]_s,$$

where $\exp(hD) = 1 + hD + \frac{h^2}{2!} D^{(2)} + \frac{h^3}{3!} D^{(3)} + \ldots$ ad inf.

It can be seen that D itself is the normal total differential operator, for if $u = u(y_1, y_2, \ldots y_n)$ is any differentiable function of the y_i, then

$$\frac{du}{dt} = \frac{\partial u.}{\partial y_1} \frac{dy_1}{dt} + \ldots + \frac{\partial u.}{\partial y_n} \frac{dy_n}{dt}$$

$$= \frac{\partial u. f_1}{\partial y_1} + \ldots + \frac{\partial u. f_n}{\partial y_n}$$

$$= \sum_j f_j u^j$$

$$= Du.$$

$D^{(2)}$, $D^{(3)}$, however, are not the same as the conventional powers D^2, D^3 defined by repeated operation, as will be seen below, and the bracketed index is used to distinguish the former.

3. THE SET OF BASIC DIFFERENTIAL OPERATORS

Consider the expansion of D^2u, where u, as before, is a function of the y_i.

$$D^2u = D(Du) = \sum_j f_j \frac{\partial}{\partial y_j} \left(\sum_k f_k \frac{\partial u}{\partial y_k} \right)$$

$$= \sum_{j,k} f_j \frac{\partial f_k}{\partial y_j} \frac{\partial u}{\partial y_k} + \sum_{j,k} f_j f_k \frac{\partial^2 u}{\partial y_j \ \partial y_k}$$

$$= \sum_{j,k} f_j f_k^j u^k + \sum_{j,k} f_j f_k u^{jk}$$

Thus, $$D^2 = \sum_{j,k} f_j f_k^j (\ )^k + D^{(2)} .$$

Similarly, it is found that

$$D^3 = \sum_{j,k,\ell} f_j f_k^j f_\ell^k (\ )^\ell + \sum_{j,k,\ell} f_j f_k f_\ell^{jk} (\ )^\ell + 3 \sum_{j,k,\ell} f_j f_k f_\ell^j (\ )^{k\ell}$$

$$+ \sum_{j,k,\ell} f_j f_k f_\ell (\ )^{jk\ell},$$

where the final term is $D^{(3)}$.

The operators with which we shall be concerned are all, in fact, generated by the normal powers of D as above.

An operator such as $f_j f_k^j f_\ell^k(\;)^\ell$ is called a SINGLE OPERATOR, the order of which (3, in this case) is the number of different indices (or the number of 'f's). It will be noticed that each single operator consists of a multiplier, which is a function of the y_i, and a partial differential operator. The n-th order operators are all generated by D^n.

Single operators are not of much interest, as the operators always occur in sets summed over all n values of each of the suffices.

Such a sum, e.g. $\sum_{j,k,\ell} f_j f_k^j f_\ell^k \;(\;)^\ell$, is called a BASIC OPERATOR and any linear combination of basic operators is called a GENERAL OPERATOR.

4. DOT PRODUCTS OF OPERATORS

Consider the effects of operating twice with $D = \sum_j f_j(\;)^j$, but regarding the multipliers f_j as being constant. It is seen that the result may be regarded as equivalent to operation with $D^{(2)}$.

This concept can be generalised and leads to what will be called the DOT PRODUCT of two operators.

Definition

If $P = p \dfrac{\partial^a}{\partial y_i \partial y_j \ldots}$ and $Q = q \dfrac{\partial^b}{\partial y_k \partial y_\ell \ldots}$, where p and q are functions of the y, then the operator $pq \dfrac{\partial^{a+b}}{\partial y_i\, \partial y_j\, \partial y_k\, \partial y_\ell \ldots}$ will be called the dot product of P and Q and will be denoted by P.Q.

The dot product of two general operators is obtained in the obvious way as the sum of the dot products of the components of the operators.

The dot product is clearly commutative, i.e. P.Q = Q.P and obeys the normal distributive and associative laws.

Dot powers, as has already been seen, are represented by putting the index in brackets, thus

$$P.P = P^{(2)}, \quad P^{(n)} . P = P^{(n+1)}.$$

The dot exponential of P is

$$\exp(P) = 1 + P + \frac{1}{2!} P^{(2)} + \frac{1}{3!} P^{(3)} + \ldots .$$

The form exp( ) is always used for the dot exponential, in order to distinguish it from the normal exponential, which is written thus:

$$e^P = 1 + P + \frac{1}{2!} P^2 + \frac{1}{3!} P^3 + \ldots .$$

An important consequence of the commutativity of the dot product, by analogy with ordinary algebra, is the exponential product law

$$\exp (P + Q) = (\exp P) \cdot (\exp Q) .$$

The normal exponential of operators does not, in general, obey this law since normal multiplication of operators is not commutative, as may be seen from Table III of normal products.

The normal exponential is encountered when considering the true values of the variables y_i of equation (1).

Thus, if $y_i(t_s)$ denotes the true value of y_i when $t = t_s$, (under some specific set of initial conditions), then

$$\begin{aligned} y_i (t_{s+1}) &= y_i (t_s + h) \\ &= y_i (t_s) + hDy_i (t_s) + \frac{1}{2!} h^2 D^2 y_i(t_s) + \ldots \\ &= e^{hD} y_i (t_s). \end{aligned}$$

5. THE PARTIAL PRODUCT OF OPERATORS

In order to complete the basic combinations of two operators which arise it is necessary to introduce yet another 'product'.

Definition

If $P = p \dfrac{\partial^a}{\partial y_i \, \partial y_j \, \cdots}$ and $Q = q \dfrac{\partial^b}{\partial y_k \, \partial y_\ell \, \cdots}$ are two single operators, with p and q functions of the y_i, then the operator

$$p \left[\frac{\partial^a}{\partial y_i \, \partial y_j \, \cdots} q \right] \frac{\partial^b}{\partial y_k \, \partial y_\ell \, \cdots} = (Pq) \frac{\partial^b}{\partial y_k \, \partial y_\ell \, \cdots}$$

will be called the PARTIAL PRODUCT of P and Q and denoted by $P \wedge Q$.

In effect, the operator Q is regarded as if its 'operator' part were constant.

The definition can be extended to general operators by the application of the normal associative and distributive laws of addition.

The partial product arises in the following way. Suppose

$$y_{i,s+1} = y_{i,s} + Pf_{i,s}$$

where P is a general operator, and $Pf_{i,s}$ means the computed value of Pf_i at the end of the s-th step.

Then $$f_{i,s+1} = f_i (y_{1,s} + Pf_{1,s}, \ldots, y_{n,s} + Pf_{n,s})$$

$$= f_{i,s} + \left[\sum_j (Pf_j) \frac{\partial}{\partial y_i} \right] f_{i,s} + \frac{1}{2!} \left[\sum_j (Pf_j) \frac{\partial}{\partial y_i} \right]^{(2)} f_{i,s} + \ldots$$

Now it is seen that

$$\sum_j (Pf_j) \frac{\partial}{\partial y_i} = P \wedge D,$$

and hence it follows that

$$f_{i,s+1} = \exp (P \wedge D) f_{i,s}.$$

Operators are distributive with respect to partial multiplication, i.e. $(P + Q) \wedge R = (P \wedge R) + (Q \wedge R)$, but are not associative, i.e. $(P \wedge Q) \wedge R \neq P \wedge (Q \wedge R)$. We do not therefore define the partial powers of an operator.

Although partial multiplication is not associative, there is one rather unusual relationship which may be classed as an associative rule, namely

$$P \wedge (Q \wedge R) = (PQ) \wedge R.$$

To prove this, it is only necessary to consider the case where R is a single operator (P and Q being general), for the result with R also general follows from the distributive laws.

Suppose then that R is a single operator, and that it can be written $R = rR'$, where r is the multiplier part of the operator.

Hence,
$$\begin{aligned} P \wedge (Q \wedge R) &= P \wedge \left\{ (Qr) R' \right\} \\ &= \left\{ P(Qr) \right\} R' \\ &= \left\{ (PQ)r \right\} R' \\ &= (PQ) \wedge R. \end{aligned}$$

6. A THEOREM CONCERNING EXPONENTIALS

There is an important relationship between the dot and normal exponentials of the operator D, namely

$$\exp \left\{ \left(\frac{e^{hD} - 1}{D} \right) \wedge D \right\} = e^{hD},$$

where h is a scalar constant.

For the purpose of proving this let us assume that we are dealing with a true solution of the equation (1).

Then, by the normal Taylor expansion about the point 's' we have

$$y_{i,s+1} = e^{hD} y_{i,s},$$

where h is the step interval.

Now let $u = u(y_1, \ldots, y_n)$ be a differentiable function of the y_i and consider its value at the end of the $s + 1$-th step.

$$u_{s+1} = u(y_{1,s+1}, \ldots, y_{n,s+1})$$

$$= u(y_{1,s} + Q_1, \ldots, y_{n,s} + Q_n)$$

where $$Q_j = (e^{hD} - 1) y_{j,s}.$$

Hence $$u_{s+1} = u_s + \left(\sum_j Q_j \frac{\partial}{\partial y_j} \right) u_s + \ldots$$

$$= \exp\left(\sum_j Q_j \frac{\partial}{\partial y_j} \right) u_s .$$

Since every term in $e^{hD} - 1$ has D as a factor, we can write

$$Q_j = \left(\frac{e^{hD} - 1}{D} \right) Dy_{j,s} = \left(\frac{e^{hD} - 1}{D} \right) f_{j,s} .$$

But also, by definition,

$$\sum_j \left\{ \left(\frac{e^{hD} - 1}{D} \right) f_{j,s} \right\} \frac{\partial}{\partial y_j} = \left(\frac{e^{hD} - 1}{D} \right) \Lambda D,$$

and hence it follows that

$$u_{s+1} = \exp\left\{ \left(\frac{e^{hD} - 1}{D} \right) \Lambda D \right\} u_s .$$

Finally, from first principles, taking the Taylor expansion of u about the point 's',

$$u_{s+1} = e^{hD} u_s ,$$

and the relationship we set out to prove follows from a comparison of these expressions.

7. A BASIC OPERATIONAL PROCEDURE

As has been mentioned before (in Section 4), expressions like $Dy_{i,s}$, $Pf_{i,s}$ do not mean the result of operating on $y_{i,s}$ or $f_{i,s}$ with D or P, but stand for $(Dy_i)_s$ and $(Pf_i)_s$.

One must avoid the pitfall of pre-multiplying equations like $y_{i,s+1} = y_{i,s} + hf_{i,s}$ by D. In fact, $f_{i,s+1} = Dy_{i,s+1} \neq Dy_{i,s} + hDf_{i,s}$ in this case.

One is often faced with the problem of determining the value of an f_i in cases like this, and in certain frequently occurring cases the answer can be written down immediately, as will now be shown.

Suppose that, in fact,

$$y_{i,s+1} = e^{hD} y_{i,s} + h^r P f_{i,s},$$

where P is some general operator.

Suppose also that u is some differentiable function of the y_i.

Then
$$u_{s+1} = u(y_{1,s+1}, \ldots, y_{n,s+1})$$
$$= u(y_{1,s} + Q_1, \ldots, y_{n,s} + Q_n)$$
$$= \exp\left(\sum_j Q_j \frac{\partial}{\partial y_j}\right) u_s,$$

where
$$Q_j = (e^{hD} - 1) y_{j,s} + h^r P f_{j,s}$$
$$= \left(\frac{e^{hD} - 1}{D}\right) f_{j,s} + h^r P f_{j,s}.$$

Substituting for Q_j in the above expression for u_{s+1}, and making use of the exponential product law for dot exponentials (Section 4) and the Theorem of Section 5, it follows that

$$u_{s+1} = \left\{ e^{hD} \circ \exp(h^r P \wedge D) \right\} u_s$$
$$= e^{hD} u_s + (h^r P \wedge D) u_s + O(h^{2r}).$$

In particular, since f_i is a function of the y_i, we have

$$f_{i,s+1} = e^{hD} f_{i,s} + (h^r P \wedge D) f_{i,s} + O(h^{2r}).$$

8. A PICTORIAL REPRESENTATION OF BASIC OPERATORS

Basic operators may be defined as the individual component sums in the expansions of D^r $(r = 1, 2, \ldots)$. The ORDER of an operator is the same as the order (r) of the generating power of D. A typical fourth order basic operator is

$$\sum_{jk\ell m} \left(f_j \frac{\partial f_k}{\partial y_j} \frac{\partial f_\ell}{\partial y_k} f_m \right) \frac{\partial^2}{\partial y_\ell \, \partial y_m}$$

This notation is too cumbersome for most purposes, and the standard (i.e. usually adopted) form, which has already been used is

$$f_j f_k^j f_\ell^k f_m \, (\ )^{\ell m}$$

in which the subscripts of the y's are used as indices, and summation over all n values of j,k,ℓ,m is understood.

The number of indices of the final, or 'operational', part (here 2) is called the DEGREE of the operator.

The general construction of such an operator of the 4-th order is as follows: write down $f_j f_k f_\ell f_m(\ \ )$, and then insert the letters j,k,l,m in this order as indices in such a way that each letter's appearance as a subscript precedes its appearance as an index.

Even the standard notation has been found to be too heavy in dealing with fourth and higher order processes, and the carrying out of the various types of operator multiplication becomes very laborious.

Let us, therefore, consider more closely the structure of the above operator by representing it in a somewhat different way. Draw a small circle (Figure 1) to represent the operand, and draw two lines from this, labelled ℓ and m, to indicate that these two letters are indices of the operand. Next, since f_m has no index, no further addition is made to the m-line. f_ℓ, however, has the index k, and so we draw another line from the end of the l-line and label it k. Similarly, since f_k has the index j, a further line is drawn from the end of the k-line and is labelled j. Finally, since f_j has no index, the picture is complete.

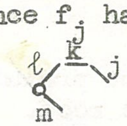


Figure 1

Figure 2

The tree representing $f_j f_k^j f_\ell^k f_m(\ \ )^{\ell m}$

It will be seen that there are just four lines representing the fact that the operator is of the fourth order, and each line has one of the four letters attached to it. Since, however, we are summing over all values of these letters, their appearance is superfluous and by omitting them, as in Figure 2, we obtain the simplest representation of the basic operator $f_j f_k^j f_\ell^k f_m(\ \ )^{\ell m}$.

Now a figure such as Figure 2 is well known in the field of topology [1,2]. It is called a TREE, or more strictly, a tree with a root, since one 'node' is singled out. A tree (Reference 1) is defined as "a connected linear graph which contains no 1-circuits". From our point of view it is reasonable to think of a tree as a connected set of freely-jointed rods such that no closed loop is formed.

The joints and the unconnected rod ends are called nodes.

If R is the number of rods (the order of the corresponding operator) and N the number of nodes then

$$N = R + 1.$$

This relation is, in fact, sufficient to distinguish trees from other connected linear graphs.

A formula for the number of trees of a given order was discovered by CAYLEY (Reference 3) and quoted by ROUSE-BALL (Reference 2). If A_r is the number of trees of order r, then

$$(1 - x)^{-1}\,(1 - x^2)^{-A_1}\,(1 - x^3)^{-A_2} \ldots = 1 + A_1 x + A_2 x^2 + A_3 x^3 + \ldots .$$

Using this formula the values of A_1, A_2, ... can be found in succession, the first seven values being

$$A_1 = 1,\ A_2 = 2,\ A_3 = 4,\ A_4 = 9,\ A_5 = 20,\ A_6 = 48,\ A_7 = 115.$$

These, then, are the number of basic operators of a given order, and the full expressions for the cases r=2=3 were given in Section 3.

A list of operators up to order 5, with their trees, and a serial notation to be explained later, is shown in Table I.

It is found that the multiplication of operators can be performed quite easily by performing operations on the corresponding trees, and the rules for doing this will now be given.

9. DOT MULTIPLICATION OF TREES

If $P = p(\ )^{\ell m \cdots}$ and $Q = q(\ )^{rs \cdots}$ represent two basic operators then their dot product is simply

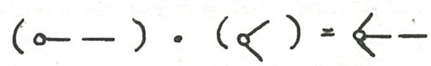
$$P.Q = pq(\ )^{\ell m \ldots rs \ldots}.$$

It is thus seen that the tree corresponding to this dot product is obtained by bringing the trees of P and Q together at the root.

As an example

$$\left[f_j f_k^j(\ )^k \right] . \left[f_\ell f_m(\ )^{\ell m} \right] = f_j f_k^j f_\ell f_m(\ )^{k \ell m}$$

and the corresponding tree equation is



The converse process can also be performed, that is to say, a tree (and hence the corresponding operator) can be factorised relative to dot multiplication by parting at the root. A tree which has only one rod stemming from its root will be called a prime tree, since it has no proper dot factors. Factorisation into prime factor trees is clearly unique, and the number of prime factor trees is equal to the number of rods stemming from the root (or the number of indicial letters of the corresponding operand), i.e. to the degree of the tree.

10. NORMAL MULTIPLICATION OF TREES

The rule for normal multiplication of trees would take too much space to derive here, so it is proposed merely to state it and give an example.

The rule is as follows.

To obtain the normal product PQ of two basic operators P and Q, first factorise P dot-wise into its prime factors, and then graft these factors by their roots at arbitrary nodes of Q, each possible placing giving a component of PQ. In this grafting, all the factors of P are to be considered as distinct.

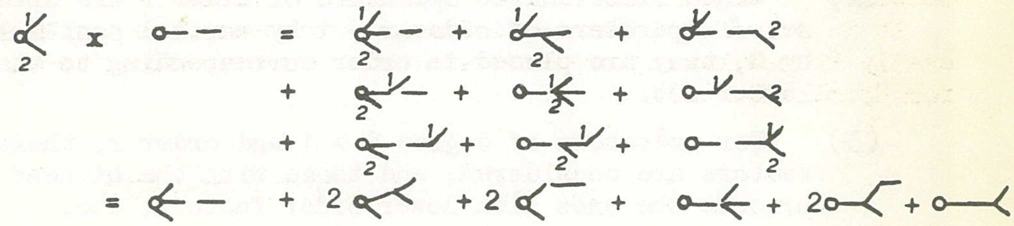
If d_p is the degree of P, and r_q is the order of Q, then it can be shown that the total number of components of PQ is $(r_q + 1)^{dp}$. This provides a useful check when actually writing out the components of a product.

If r_p is the order of P, then the order of PQ is $r_p + r_q$.

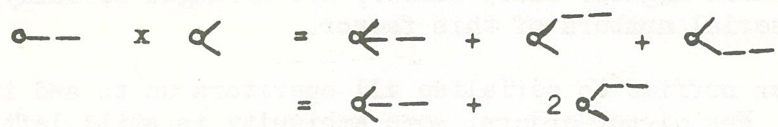
Example $\quad P = f_j f_k(\ )^{jk}, \quad Q = f_j f_k^j(\ )^k$.

Here, $r_p = d_p = r_q = 2$, $d_q = 1$, so that the normal product PQ will have 9 components, whilst the product QP will have only 3.

For PQ we have



and for QP we have



11. PARTIAL MULTIPLICATION OF TREES

By studying the definition of a partial product it is seen that the partial product $P \wedge Q$ of the basic operators P and Q consists of those components of the normal product PQ which have the same degree as Q.

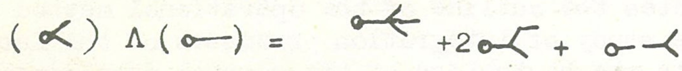
It follows that the rule for multiplying the corresponding trees is as follows.

Separate P into its prime factor trees, and then graft these factors by their roots at arbitrary nodes of Q except at its root, each possible placing giving a component of $P \wedge Q$. In this grafting, all the factors of P are to be considered as distinct.

Since the number of nodes of Q upon which the grafting takes place is r_q (compared with $r_q + 1$ in the case of normal multiplication) the total number of components of $P \wedge Q$ is $r_q^{d_p}$.

Example

Taking the same basic operators P and Q as in Section 10 above, we have



and for $Q \wedge P$ we have

12. SERIALISATION OF BASIC OPERATORS

Although trees have proved to be a great help in obtaining basic products and in studying the structure of operators, they are not very convenient in practical work. A further notation is therefore proposed, in which the basic operators or trees of each order are numbered serially.

Theletter D is used with two indices in brackets, e.g. $D^{(5,17)}$, the first index indicating the order of the operator and the second index is a serial number within that order. There are, of course, a large number of ways in which the serial numbers might be allotted to the operators but the way which has been adopted seems the most natural (Table I).

The rules for allocating serial numbers are as follows.

(1) Operators are arranged in ascending order by their degrees.

(2) Since first degree operators of order r are obtained from the set of operators of order r − 1 by partial post-multiplication by D, they are placed in order corresponding to the order of this latter set.

(3) For operators of degree d > 1 and order r, their dot-wise factors are considered, and those with the highest order factor precede the ones with lower order factors, etc.

(4) Operators of degree d > 1 and order r, and having the same order for their highest order factor, are arranged serially according to the serial numbers of this factor.

These rules suffice to serialise all operators up to and including the 6-th order. For higher orders, some ambiguity is still left, but this problem has not yet been considered.

In each order the final operator is found to be the corresponding dot-power of D, so that the serial number may be omitted in this case and we would use e.g. $D^{(4)}$ and not $D^{(4,9)}$.

Multiplication tables for basic operators have been worked out using trees and then translated into the serial notation. The results are given in Tables II, III and IV.

As pointed out already, the basic operators of the r-th order are all contained in the expansion of D^r. A list of the coefficients of the basic operators in these expansions is given in Table I. For example

$$D^3 = D^{(3,1)} + D^{(3,2)} + 3D^{(3,3)} + D^{(3)} .$$

To post-multiply an operator partially by D, it is only necessary to add 1 to the order index of each component, taking care to insert the serial number in any dot-power which is present.

For example, $$D^3 \wedge D = D^{(4,1)} + D^{(4,2)} + 3D^{(4,3)} + D^{(4,4)} .$$

This completes the outline of the operational method in so far as it relates to the study of integration processes of the Kutta type, but only represents the beginnings of the general development which is possible.

As an example of the method, a certain 5-stage Kutta process is considered in an appendix to this paper.

Guided Weapons Dept.,
RAE, Farnborough, Hants

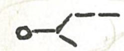
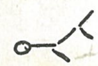
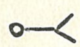
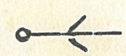
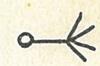
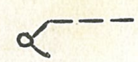
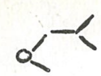
25th April, 1957.

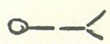
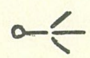
REFERENCES

No.	Author	Title
1	Veblen, O.	Analysis Situs. Amer. Math. Soc. 1931, p. 18
2	Rouse Ball, W.W.	Mathematical Recreations and Essays, Macmillan, London
3	Cayley	Philosophical Magazine, March, 1857, pp. 172-6
4	Cayley	Quarterly Journal of Mathematics London, 1889 vol. XXIII, pp. 376-8.

TABLE I

NOTATIONS FOR BASIC OPERATORS

Standard Notation	Tree	Serial Notation		Standard Notation	Tree	Serial Notation	Coeff in D^5
$f_i(\,)^i$		$D = D^{(1)}$		$f_i f^i_j f^j_k f^k_\ell f^\ell_m(\,)^m$		$D^{(5,1)}$	1
			coeff in D^2	$f_i f_j f^{ij}_k f^k_\ell f^\ell_m(\,)^m$		$D^{(5,2)}$	1
$f_i f^i_j(\,)^j$		$D^{(2,1)}$	1	$f_i f^i_j f_k f^{jk}_\ell f^\ell_m(\,)^m$		$D^{(5,3)}$	3
$f_i f_j(\,)^{ij}$		$D^{(2)}$	1	$f_i f_j f_k f^{ijk}_\ell f^\ell_m(\,)^m$		$D^{(5,4)}$	1
			coeff in D^3	$f_i f^i_j f^j_k f_\ell f^{k\ell}_m(\,)^m$		$D^{(5,5)}$	4
$f_i f^i_j f^j_k(\,)^k$		$D^{(3,1)}$	1	$f_i f_j f^{ij}_k f_\ell f^{k\ell}_m(\,)^m$		$D^{(5,6)}$	4
$f_i f_j f^{ij}_k(\,)^k$		$D^{(3,2)}$	1	$f_i f_j f^i_k f^j_\ell f^{k\ell}_m(\,)^m$		$D^{(5,7)}$	3
$f_i f^i_j f_k(\,)^{jk}$		$D^{(3,3)}$	3	$f_i f^i_j f_k f_\ell f^{jk\ell}_m(\,)^m$		$D^{(5,8)}$	6
$f_i f_j f_k(\,)^{ijk}$		$D^{(3)}$	1	$f_i f_j f_k f_\ell f^{ijk\ell}_m(\,)^m$		$D^{(5,9)}$	1
			coeff in D^4	$f_i f^i_j f^j_k f^k_\ell f_m(\,)^{\ell m}$		$D^{(5,10)}$	5
$f_i f^i_j f^j_k f^k_\ell(\,)^\ell$		$D^{(4,1)}$	1	$f_i f_j f^{ij}_k f^k_\ell f_m(\,)^{\ell m}$		$D^{(5,11)}$	5

Standard Notation	Tree	Serial Notation	
$f_i f_j f^{ij}_k f^k_\ell ()^\ell$		$D^{(4,2)}$	1
$f_i f^i_j f_k f^{jk}_\ell ()^\ell$		$D^{(4,3)}$	3
$f_i f_j f_k f^{ijk}_\ell ()^\ell$		$D^{(4,4)}$	1
$f_i f^i_j f^j_k f_\ell ()^{k\ell}$		$D^{(4,5)}$	4
$f_i f_j f^{ij}_k f_\ell ()^{k\ell}$		$D^{(4,6)}$	4
$f_i f_j f^i_k f^j_\ell ()^{k\ell}$		$D^{(4,7)}$	3
$f_i f^i_j f_k f_\ell ()^{jk\ell}$		$D^{(4,8)}$	6
$f_i f_j f_k f_\ell ()^{ijk\ell}$		$D^{(4)}$	1

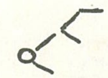
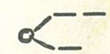
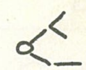
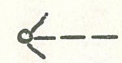
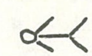
Standard Notation	Tree	Serial Notation	Coeff in D^5
$f_i f^i_j f_k f^{jk}_\ell f_m ()^{\ell m}$		$D^{(5,12)}$	15
$f_i f_j f_k f^{ijk}_\ell f_m ()^{\ell m}$		$D^{(5,13)}$	5
$f_i f^i_j f_k f^j_\ell f^k_m ()^{\ell m}$		$D^{(5,14)}$	10
$f_i f_j f_k f^{ij}_\ell f^k_m ()^{\ell m}$		$D^{(5,15)}$	10
$f_i f^i_j f^j_k f_\ell f_m ()^{k\ell m}$		$D^{(5,16)}$	10
$f_i f_j f^{ij}_k f_\ell f_m ()^{k\ell m}$		$D^{(5,17)}$	10
$f_i f_j f^i_k f^j_\ell f_m ()^{k\ell m}$		$D^{(5,18)}$	15
$f_i f^i_j f_k f_\ell f_m ()^{jk\ell m}$		$D^{(5,19)}$	10
$f_i f_j f_k f_\ell f_m ()^{ijk\ell m}$		$D^{(5)}$	1

TABLE II

DOT PRODUCTS OF BASIC OPERATORS

Note : Dot products are commutative.

$D \cdot D = D^{(2)}$	$D \cdot D^{(4,1)} = D^{(5,10)}$	$D^{(2,1)} \cdot D^{(3,1)} = D^{(5,14)}$
	$D \cdot D^{(4,2)} = D^{(5,11)}$	$D^{(2,1)} \cdot D^{(3,2)} = D^{(5,15)}$
$D \cdot D^{(2,1)} = D^{(3,3)}$	$D \cdot D^{(4,3)} = D^{(5,12)}$	$D^{(2,1)} \cdot D^{(3,3)} = D^{(5,18)}$
$D \cdot D^{(2)} = D^{(3)}$	$D \cdot D^{(4,4)} = D^{(5,13)}$	$D^{(2,1)} \cdot D^{(3)} = D^{(5,19)}$
	$D \cdot D^{(4,5)} = D^{(5,16)}$	
$D \cdot D^{(3,1)} = D^{(4,5)}$	$D \cdot D^{(4,6)} = D^{(5,17)}$	$D^{(2)} \cdot D^{(3,1)} = D^{(5,16)}$
$D \cdot D^{(3,2)} = D^{(4,6)}$	$D \cdot D^{(4,7)} = D^{(5,18)}$	$D^{(2)} \cdot D^{(3,2)} = D^{(5,17)}$
$D \cdot D^{(3,3)} = D^{(4,8)}$	$D \cdot D^{(4,8)} = D^{(5,19)}$	$D^{(2)} \cdot D^{(3,3)} = D^{(5,19)}$
$D \cdot D^{(3)} = D^{(4)}$	$D \cdot D^{(4)} = D^{(5)}$	$D^{(2)} \cdot D^{(3)} = D^{(5)}$
$D^{(2,1)} \cdot D^{(2,1)} = D^{(4,7)}$		
$D^{(2,1)} \cdot D^{(2)} = D^{(4,8)}$		
$D^{(2)} \cdot D^{(2)} = D^{(4)}$		

TABLE III

NORMAL PRODUCTS OF BASIC OPERATORS

$$D \times D = D^2 = D^{(2,1)} + D^{(2)}$$

$$D \times D^{(2,1)} = D^{(3,1)} + D^{(3,2)} + D^{(3,3)}$$

$$D \times D^{(2)} = 2D^{(3,3)} + D^{(3)}$$

$$D \times D^{(3,1)} = D^{(4,1)} + D^{(4,2)} + D^{(4,3)} + D^{(4,5)}$$

$$D \times D^{(3,2)} = 2D^{(4,3)} + D^{(4,4)} + D^{(4,6)}$$

$$D \times D^{(3,3)} = D^{(4,5)} + D^{(4,6)} + D^{(4,7)} + D^{(4,8)}$$

$$D \times D^{(3)} = 3D^{(4,8)} + D^{(4)}$$

$$D^{(2,1)} \times D^{(2,1)} = D^{(4,1)} + D^{(4,3)} + D^{(4,7)}$$

$$D^{(2)} \times D^{(2)} = 2D^{(4,6)} + 2D^{(4,7)} + 4D^{(4,8)} + D^{(4)}$$

$$D^{(2,1)} \times D^{(2)} = 2D^{(4,5)} + D^{(4,8)}$$

$$D^{(2,1)} \times D = D^{(3,1)} + D^{(3,3)}$$

$$D^{(2)} \times D = D^{(3,2)} + 2D^{(3,3)} + D^{(3)}$$

$$D^{(3,1)} \times D = D^{(4,1)} + D^{(4,5)}$$

$$D^{(3,2)} \times D = D^{(4,2)} + D^{(4,6)}$$

$$D^{(3,3)} \times D = D^{(4,3)} + D^{(4,5)} + D^{(4,7)} + D^{(4,8)}$$

$$D^{(3)} \times D = D^{(4,4)} + 3D^{(4,6)} + 3D^{(4,8)} + D^{(4)}$$

$$D^{(2)} \times D^{(2,1)} = D^{(4,2)} + 2D^{(4,3)} + D^{(4,4)} + 2D^{(4,5)} + 2D^{(4,6)} + D^{(4,8)}$$

TABLE IV

PARTIAL PRODUCTS OF BASIC OPERATORS

$D \wedge D = D^{(2,1)}$

$D \wedge D^{(2,1)} = D^{(3,1)} + D^{(3,2)}$

$D \wedge D^{(2)} = 2D^{(3,3)}$

$D \wedge D^{(3,1)} = D^{(4,1)} + D^{(4,2)} + D^{(4,3)}$

$D \wedge D^{(3,2)} = 2D^{(4,3)} + D^{(4,4)}$

$D \wedge D^{(3,3)} = D^{(4,5)} + D^{(4,6)} + D^{(4,7)}$

$D \wedge D^{(3)} = 3D^{(4,8)}$

$D^{(2,1)} \wedge D^{(2,1)} = D^{(4,1)} + D^{(4,3)}$

$D^{(2)} \wedge D^{(2)} = 2D^{(4,6)} + 2D^{(4,7)}$

$D^{(2,1)} \wedge D^{(2)} = 2D^{(4,5)}$

$D^{(2,1)} \wedge D = D^{(3,1)}$

$D^{(2)} \wedge D = D^{(3,2)}$

$D^{(3,1)} \wedge D = D^{(4,1)}$

$D^{(3,2)} \wedge D = D^{(4,2)}$

$D^{(3,3)} \wedge D = D^{(4,3)}$

$D^{(3)} \wedge D = D^{(4,4)}$

$D^{(2)} \wedge D^{(2,1)} = D^{(4,2)} + 2D^{(4,3)} + D^{(4,4)}$

APPENDIX

A 5-STAGE, 4-TH ORDER KUTTA PROCESS

An integration process was required having the following characteristics :-

(1) No special starting routine (hence a Kutta process is indicated).

(2) At each stage the variables should be accurate to as high an order as possible.

(3) The two final stages should relate to the end of the step.

(4) The elementary truncation error should be of the fifth order in h, the interval of integration.

It is found that to satisfy these characteristics a 5-stage process is required, and the general form may be written

$$
\begin{aligned}
y_1 &= y_o + haf_o \\
y_2 &= y_o + hbf_o + hcf_1 \\
y_3 &= y_o + hdf_o + hgf_2 \\
y_4 &= y_o + hkf_o + hlf_2 + hmf_3 \\
y_5 &= y_o + hpf_o + hqf_3 + hrf_4
\end{aligned}
$$

where y, f relate to any of the n variables y_i (of equation 1) and the suffix here relates to the stage number.

A preliminary study showed the necessity for omitting terms like 'f_1' in the expression for y_3 in order to satisfy characteristic (2).

In fact, the error in y_1 will be of order h^2,
that of y_2 will be of order h^3,
that of y_3, y_4 will be of order h^4,
and that of y_5 will be of order h^5,

Let

$$
\boxed{\begin{aligned}
b + c &= u \\
d + g &= v \\
k + \ell + m &= 1 \\
p + q + r &= 1
\end{aligned}}
$$

and also (the last two equations)

the latter pair being required to satisfy characteristic (3).

Then, applying the principles outlined in the text, we have, first

$$
\begin{aligned}
f_1 &= \exp(haD)f_o \\
&= \left(1 + haD + \frac{h^2}{2} a^2 D^{(2)} + \frac{h^3}{6} a^3 D^{(3)}\right) f_o + O(h^4).
\end{aligned}
$$

Hence, $y_2 = y_o + huf_o + h^2 caDf_o + \frac{h^3}{2} ca^2 D^{(2)} f_o + \frac{h^4}{6} ca^3 D^{(3)} f_o + O(h^5)$.

Comparison with $e^{huD} y_o$ up to the 2-nd order in h gives

$$\boxed{u^2 = 2ca}$$

and $$y_2 = e^{huD} y_o + h^3 P_1 f_o + h^4 P_2 f_o + O(h^5)$$

where $$P_1 = \frac{ca^2}{2} D^{(2)} - \frac{u^3}{6} D^2, \text{ and } P_2 = \frac{ca^3}{6} D^{(3)} - \frac{u^4}{24} D^3 .$$

Using the procedure of section 6 we then have

$$f_2 = e^{huD} f_o + h^3 (P_1 + hP_2) \Lambda Df_o + O(h^5),$$

and hence

$$y_3 = y_o + hdf_o + hge^{huD} f_o + h^4 g (P_1 + hP_2) \Lambda Df_o + O(h^6)$$

$$= y_o + hvf_o + h^2 guDf_o + \frac{h^3}{2} gu^2 D^2 f_o + \frac{h^4}{6} gu^3 D^3 f_o + h^4 gP_1 \Lambda Df_o + O(h^5).$$

Comparison with e^{hvD} y up to the 3-rd order in h gives

$$\boxed{\begin{aligned} v^2 &= 2gu \\ v^3 &= 3gu^2 \end{aligned}}$$

and $$y_3 = e^{hvD} y_o + h^4 Qf_o + O(h^5),$$

where $$Q = \frac{gu^3}{6} D^3 + gP_1 \Lambda D - \frac{v^4}{24} D^3 .$$

It then follows that

$$f_3 = e^{hvD} f_o + h^4 Q \Lambda Df_o + O(h^5),$$

and so

$$y_4 = y_o + hkf_o + hle^{huD} f_o + h^4 \ell (P_1 + hP_2) \Lambda Df_o + hme^{hvD} f_o + h^5 mQ \Lambda Df_o + O(h^6).$$

$$= y_o + hf_o + h^2(lu + mv) Df_o + \frac{h^3}{2}(lu^2 + mv^2) D^2 f_o + \frac{h^4}{6} (lu^3 + mv^3) D^3 f_o + \frac{h^5}{24} (lu^4 + mv^4) D^4 f_o + h^4 \ell (P_1 + hP_2) \Lambda Df_o + h^5 mQ \Lambda Df_o + O(h^6).$$

Comparison with e^{hD} up to the 3-rd order in h gives

$$lu + mv = \tfrac{1}{2}$$
$$lu^2 + mv^2 = \tfrac{1}{3}$$

and $$y_4 = e^{hD} y_o + h^4 R_1 f_o + h^5 R_2 f_o + O(h^6),$$

where $$R_1 = \frac{1}{6}(lu^3 + mv^3)D^3 + \ell P_1 \Lambda D - \frac{1}{24} D^3$$

and $$R_2 = \frac{1}{24}(lu^4 + mv^4)D^4 + mQ\Lambda D - \frac{1}{120} D^4$$

It follows that

$$f_4 = e^{hD} f_o + h^4 (R_1 + hR_2)\Lambda Df_o + O(h^6),$$

and so $$y_5 = y_o + hpf_o + hqe^{hvD} f_o + h^5 qQ\Lambda Df_o + hre^{hD} f_o + h^5 rR_1 \Lambda Df_o + O(h^6)$$

$$= y_o + hf_o + h^2(qv + r)Df_o + \tfrac{1}{2}h^3(qv^2 + r)D^2 f_o$$

$$+ \frac{1}{6}h^4(qv^3 + r)D^3 f_o + \frac{1}{24}h^5(qv^4 + r)D^4 f_o + h^5(qQ + rR_1)\Lambda Df_o$$

$$+ O(h^6).$$

Comparison with e^{hD} up to the 4-th order in h gives

$$\boxed{\begin{aligned} qv + r &= \tfrac{1}{2} \\ qv^2 + r &= \tfrac{1}{3} \\ qv^3 + r &= \tfrac{1}{4} \end{aligned}}$$

and $$y_5 = e^{hD} y_o + h^5 Sf_o + O(h^6),$$

where $$S = \frac{1}{24}(qv^4 + r)D^4 + (qQ + rR_1)\Lambda D - \frac{1}{120} D^4 .$$

Solution of the equations in blocks gives

$v = \tfrac{1}{2}$, $q = {}^2/_3$, $r = {}^1/_6$, $p = {}^1/_6$, $u = \tfrac{1}{3}$, $m = 2$, $\ell = -{}^3/_2$, $k = \tfrac{1}{2}$, $g = \tfrac{3}{8}$, $d = \tfrac{1}{3}$,

and with a one-parameter ambiguity, $ca = {}^1/_{18}$, $b = \tfrac{1}{3} - \frac{1}{18a}$.

Substituting these values in the "truncation" operators, and expanding in terms of basic operators, it is found that

$$R_1 = -\frac{a}{24} D^{(3,2)} - \frac{1}{36} D^{(3,3)} - \frac{1}{108} D^{(3)} \qquad (D^{(3,1)} \text{ is absent}),$$

$$R_2 = -\frac{1}{120} D^{(4,1)} + \left(\frac{a}{48} - \frac{1}{120}\right) D^{(4,2)} - \frac{1}{90} D^{(4,3)} - \left(\frac{a^2}{72} + \frac{1}{90}\right) D^{(4,4)}.$$

$$- \frac{101}{25920} \left(4 D^{(4,5)} + 4 D^{(4,6)} + 3 D^{(4,7)} + 6 D^{(4,8)} + D^{(4)}\right),$$

and

$$S = \frac{1}{2880} D^4 - \frac{1}{576} D^3 \Lambda D$$

$$= -\frac{1}{720} D^{(4,1)} - \frac{1}{720} D^{(4,2)} - \frac{1}{240} D^{(4,3)} - \frac{1}{720} D^{(4,4)}$$

$$+ \frac{1}{2880} \left(4 D^{(4,5)} + 4 D^{(4,6)} + 3 D^{(4,7)} + 6 D^{(4,8)} + D^{(4)}\right).$$

The value of a can be chosen to suit the convenience of programming, and $a = \frac{1}{3}$ is suggested. In this case the process equations are

$$y_1 = y_o + \tfrac{1}{3} hf_o$$

$$y_2 = y_o + \frac{1}{6} hf_o + \frac{1}{6} hf_1$$

$$y_3 = y_o + \tfrac{1}{8} hf_o + \tfrac{3}{8} hf_2$$

$$y_4 = y_o + \tfrac{1}{2} hf_o - {}^3/_2\, hf_2 + 2hf_3$$

$$y_5 = y_o + {}^1/_6\, hf_o + {}^2/_3\, hf_3 + {}^1/_6\, hf_4 \,.$$

If the differential equations are linear with constant coefficients, then the truncation errors of the 4-th and 5-th stages reduce to

$$y_4 - e^{hD} y_o = -\frac{1}{120} h^5 y_o(v) + 0(h^6)$$

and

$$y_5 - e^{hD} y_o = -\frac{1}{720} h^5 y_o^{(v)} + 0(h^6),$$

so that practical estimates of the errors introduced at each step may be obtained in the usual way.

DISCUSSION

Dr. S. Gill, Ferranti Ltd.

I looked into this subject 5 years ago but I did not carry the subject quite as far as Mr. Merson for two reasons.

Firstly, automatic computing was new then and there did not seem to be much interest in a variety of methods for integrating differential equations. Since then it has become apparent that there are applications for a number of different processes and perhaps it is now worthwhile standardising the procedures for developing variations of the process.

The other reason is that, not being a pure mathematician I was never quite sure of what I was talking about. It is difficult to keep a cool head when discussing the various derivatives of one variable with respect to another and I think this would justify a solid attack by a pure mathematician to put everything on a sound basis. I did however see the one-one correspondence between the trees and the "basic operators" and I embarked on a project for programming EDSAC to set up the equations to be satisfied by any new process. A tree was to be represented by a suitable binary word, which was obtained by skirting round the tree in a clockwise direction and registering a one every time one turns to the right and a zero every time one turns to the left. It was necessary to devise a routine to reduce all trees to a standard form, since different binary numbers could represent the same operator. I chose the word which was numerically greater. The writing of a routine to produce such a standard form was as far as I went.

Professor T.M. Cherry, University of Melbourne.

I should be glad if something could be said concerning the circumstances in which different formulae are either more or less desirable. I am new to this subject and would be glad of information.

Mr. R.H. Merson (In Reply)

[I think that as far as automatic computers are concerned difference methods are not very good as usually a lot of iteration is required. They have been used but I feel that R. Kutta process is better because no starting routine is required and the programming is more straight forward. There is no basic reason why one should be better than another. I think it depends on the use to which one is putting it.] Some information on this should come from Dr. Wilkes' paper.

Dr. J.M. Bennett, University of Sydney.

I have great respect for the R. Kutta process as a means of starting off the integration of simultaneous d.e.'s in the way you have described and also on occasions where you have a certain amount of random noise injected into the system, in which case there is little alternative but to treat each interval individually. But surely the predictor-corrector techniques (Milnes for example) will result in one getting an answer to the same accuracy in a shorter time simply because information outside a single interval is used.

Some work, which seems to have been neglected, (carried out in 1925, I believe) extends the R. Kutta type process to simultaneous second order differential equations. The work was by Nyström published in Acta Soc. Sci. Fennicae; I should like to know if you have seen this article and whether you have considered applying this symbolic treatment for deriving Kutta type processes for higher order equations. I once played with the type of expansion you used and found that the formula derived by Nyström could be more easily derived algebraically.

Mr. R.H. Merson (In Reply)

In answer to Dr. Bennett's first question, I have found that the predictor-corrector formulae do not have the wide stability range of R. Kutta processes. I am interested in dynamic problems and I find that you can use with the R. Kutta process an integration interval perhaps seven times as long as with the Milne-predictor. In fact the R. Kutta is more economical.

Dr. J.M. Bennett, University of Sydney.

Is the figure of seven for the same theoretical truncation error?

Mr. R.H. Merson (In Reply)

For the same overall accuracy.

Dr. J.M. Bennett, University of Sydney.

Yes, but you can choose formulae of both types so that the theoretical truncation error is of any order you wish. Are you comparing formulae of the same theoretical truncation error?

Mr. R.H. Merson (In Reply)

I am talking about the stability of the formula not just the truncation error. Whatever the truncation error, when solving say a set of simultaneous linear d.e.'s with any process there is an interval beyond which the process diverges.

Dr. J.M. Bennett, University of Sydney.

And this is a question of choosing a predictor-corrector formula such that the solution of the difference equation does not contain unstable terms.

Mr. R.H. Merson (In Reply)

Yes! But even so, I think all of the predictor-corrector formulae of the open type (i.e. without an infinite set of iterations) will be unstable for a sufficiently large integration interval.

Dr. J.M. Bennett, University of Sydney.

I think in M.T.A.C. about a year ago someone published an example showing that if you go beyond the first corrector you will be led to instability that would not otherwise occur. This is a question of examining the formula you are using. I am surprised at the figure of seven.

Professor T.M. Cherry, University of Melbourne.

I think the essential point is the maximum amount of information one can derive from the number of function values calculated. The predictor-corrector would require, say, two function values for each interval whereas the R. Kutta process requires four function values.

I think the significant quantity is the interval covered for each function value calculated, always within a given accuracy.

Mr. R.H. Merson (In Reply)

Yes. That is the sort of thing I have used to assess the relative merits of processes.

In reply to Dr. Bennett's second question I have not seen the paper by Nystrom. Was it in English?

Dr. J.M. Bennett, University of Sydney.

In German actually, not Finnish.

Mr. R.H. Merson

I did mention that I have worked out a 5 stage 4th order process which has been programmed for the Pegasus computer and it is twice as fast as the process of integrating normally over one step and two half steps. (See p. 17 of paper).

Dr. A.S. Douglas, University of Leeds.

I think that an important point has been left out on the question of R. Kutta versus Milne Predictor-Corrector, and this is the important role played by change of step length. One of the major advantages of R. Kutta is that you can change step length readily at any time, whereas the other method gives a lot more trouble although quite possible.

Dr. J.M. Bennett, University of Sydney.

It means changing 3 coefficients in the usual Predictor Corrector formula which is no great hardship.

Mr. J.H. Wilkinson, National Physical Laboratory.

When the R.K. process was first used on digital computers the criticism that skilled computers on desk machines used against it was that you had a fixed order formula and you had to reduce interval size to compensate for this, whereas on desk machines a skilled operator had not necessarily decided on the order of the formula in advance and they chose a convenient interval for good progress and then took enough differences to get suitable accuracy. This criticism has some justification but in my experience a much more common weakness has been associated with the stability problem.

In solving a set of linear d.e.'s with say, for simplicity, constant coefficients and one has solutions like e^{-t} and e^{-50t}, the e^{-50t} often gives trouble and forces you to take very small intervals, whereas in an implicit method one can use a formula with a much higher truncation error than the R.Kutta process and yet carry out an accurate integration over a very much larger interval.

Mr. R.H. Merson (In Reply)

I have tried to combine a R. Kutta process with another type of process for use when one has high frequency transients in the system. You cannot combine R. Kutta or Kutta-Gill processes with other processes because the order of accuracy of successive stages does not increase monotonically. This is hard to see but is true. In this process which I have, one of the specifications was that every step should be one order better in accuracy than the preceding one and that would enable one to combine this with other processes. It took me a long time to see this but the explanation is hard to put in writing.

I have in fact solved a set of equations which had previously been used as a test set up on T.R.I.D.A.C. I used a method which combined a second order Kutta process with a similar process in which the coefficients were functions of the coefficients of the equations to be solved. I found this overcame the difficulty caused by the high frequency term. The short interval being used until the h.f. term died out and after this the interval was successfully lengthened.