Runge–Kutta Starters for Multistep Methods

C. W. GEAR
University of Illinois

Runge-Kutta-like formulas which enable a multistep method to start or restart at a high order after just one Runge-Kutta (RK) step are presented. These formulas greatly improve the efficiency of multistep methods in a situation in which they were previously out performed by RK methods—i e., in which there are problems with frequent discontinuities or sudden large increases in derivatives, all of which cause an automatic program to reduce order and step size suddenly.

Key Words and Phrases: Runge-Kutta, multistep, integration, ordinary differential equations
CR Categories: 5.17

1. INTRODUCTION

Twenty years ago discussions of the relative merits of multistep versus Runge-Kutta (RK) methods made frequent references to the "difficulty" of starting a multistep method, that is, the cost of obtaining the additional $k - 1$ values needed to start a k-step method when just the initial value for the problem is available. Some early programs for constant-step, constant-order multistep methods used several RK steps to start. For example, a four-step Adams formula can be started with three fourth-order RK steps and a final function evaluation to obtain y'_i, $i = 0, 1, 2$, and 3, and y_3 from y_0. No error control was provided in such a program, but at that time the error control in most programs consisted of an external adjustment of the step by the user. Nordsieck [12] designed one of the first multistep methods with automatic starting, but the introduction of variable-order methods finally "solved" the problem because these methods can start at whichever order corresponds to a one-step method—usually first or second order (see, for example, Gear [6], Hindmarsh [9], Krogh [11], and Shampine and Gordon [13]). It is interesting to observe that Nordsieck's starting scheme, which consisted of integrating forward and backward over several steps several times, amounts to a variable-order scheme: because Adams methods are used and because of the representation used, it can be shown that the order is increased at each step in the Nordsieck starter.

This work was supported in part by the U.S Department of Energy under Contract US ENERGY/EY-76-S-02-2383.
Author's address: Department of Computer Science, University of Illinois, Urbana, IL 61801.

Starting at first or second order and relying on the order control to increase the order to an appropriate value is programmatically simple because the mechanism is already present in the program. It also has the advantage of providing local error control, which was not really present in the Nordsieck scheme. However, it is not very efficient for any but low accuracies as very small step sizes must be taken at low orders to maintain accuracy. This inefficiency is often offset by the improved efficiency of high-order multistep methods for many classes of problems (see Hull et al. [10]). However, there are large classes of problems, arising, for example, in simulation, in which there are frequent discontinuities in first or higher derivatives. Since multistep methods are based on polynomial interpolations over an interval of several steps, they break down when the interval includes a discontinuity in a derivative of degree lower than the polynomial. Hence a restart is necessary. One-step methods such as RK do not have this difficulty as long as a mesh point falls exactly on each discontinuity, and, therefore, RK methods are usually superior for these "nonsmooth" problems.

This paper presents an RK-like technique for starting (or restarting) any of the currently popular automatic multistep ODE integrators at a high order. (An order 4 starter is recommended for general use.) The technique uses fewer function evaluations than are used when a conventional RK method is applied repeatedly, and it provides some degree of error control as well as a good estimate of the step size to be used in the first step of the multistep method.

Section 2 presents the idea in a non-RK framework to establish the existence of methods of the desired type and to derive an upper bound on the number of function evaluations needed in an explicit RK starter ($1 + p(p-1)/2$ for a pth-order method). Section 3 examines the lower bound on the number of function evaluations in explicit RK starters for $p \leq 4$ and suggests a particular fourth-order method that uses six function evaluations (the lower bound). Section 4 examines implicit RK starters which might be useful for some stiff problems and proves that the minimum number of stages for a pth-order starter is p and that p-stage methods with desirable stability properties exist. However, most problems are not stiff immediately after a discontinuity because the fast transients are dominant. (It is important to remember that a problem is stiff *only* when the fastest components are negligible. In the transient region a nonstiff integrator should be used.) Therefore, implicit methods are not generally recommended; Section 4 is present mainly for completeness. Section 5 discusses error estimation and step control, while the final section presents some numerical test results.

2. STARTING BY EXTRAPOLATION

If a k-value multistep method is to be started at pth order, it is necessary to generate $k-1$ additional starting values, and it is desirable that these be accurate to the $(p-1)$st or pth order. In many codes k is $p+1$ because an explicit pth-order predictor uses $p+1$ values. We consider this case and examine techniques which generate p additional pth-order accurate starting values. Some codes store values of the solution y and/or its derivative y' at a set of $p+1$ mesh points, some store backward differences, while others use the Nordsieck vector of scaled derivatives $h^j y^{(j)}/j!$. This is not important to our discussion because it is possible to compute any one set from another to pth-order accuracy. If we can generate $h^j y^{(j)}, j = 0, \ldots, p$, with error no greater than $O(h^{p+1})$, we can easily compute the starting values for any code. Our objective is to look for an explicit RK-like process of the form

$$\begin{aligned} k_i &= hf(y_0 + \sum_{j=1}^{i-1} \beta_{i-1j} k_j), \qquad i = 1, \ldots, q, \\ h^s y^{(s)} &= \sum_{j=1}^{q} k_j \gamma_{js} + O(h^{p+1}), \qquad s = 1, \ldots, p. \end{aligned} \tag{2.1}$$

In this section we give an extrapolation technique for finding coefficients which satisfy (2.1) for which $q = 1 + p(p-1)/2$.

Let y_i^m, $i = 0, \ldots, m$, be the results obtained from the application of the forward Euler method for m steps to the differential equation $y' = f(y, t)$, $y(0) = y_0$, using step size $h_m = H/m$. We assume that f has as many partial derivatives as needed. (For example, bounded fourth partial derivatives are needed for the fourth-order method.) Then it is known that there exists an asymptotic error expansion of the form

$$y_i^m = y(ih_m) + \sum_{q=1}^{p} h^q e_q(ih_m) + O(H^{p+1}), \tag{2.2}$$

where $e_q(t)$ are functions that satisfy certain differential equations and have a number of bounded derivatives. (Details can be found in Stetter [15], but are not important to our discussion. All we need to know is that this expansion exists and that the e_q are differentiable.) It follows from trivial algebra that there exist numerical differentiation formulas from $m + 1$ equally spaced points of the form

$$h_m^k z^{(k)}\left(\frac{H}{2}\right) = \sum_{i=0}^{m} d_{ik} z(ih_m) + \sum_{s=k+1}^{p} c_{sk} h_m^s z^{(s)}\left(\frac{H}{2}\right) + O(H^{p+1}) \tag{2.3}$$

for $m \geq k$, where z is any function with $p + 1$ bounded derivatives. The d_{ik} are the differentiation formula coefficients and the c_{sk} are the coefficients of the error terms. Define

$$D_m^k = \sum_{i=0}^{m} d_{ik} y_i^m. \tag{2.4}$$

Substituting (2.2) into (2.4) with $k = m = p$, applying (2.3) to resulting terms of the form $\sum_{i=0}^{p} d_{ip} z(ih_p)$ with z equal to y and e_q, and dropping terms of order $O(H^{p+1})$, we find that

$$D_p^p = h_p^p y^{(p)}\left(\frac{H}{2}\right) + O(H^{p+1}). \tag{2.5}$$

Since the left-hand side of (2.5) can be calculated, we can form an asymptotically correct value for $H^p y^{(p)}(H/2)$.

Next we examine D_{p-1}^{p-1} and D_p^{p-1}. By making the same substitutions, we arrive at the relations

$$D_{p-1}^{p-1} = h_{p-1}^{p-1} y^{(p-1)} + h_{p-1}^{p} e_1^{(p-1)} - c_{pp} h_{p-1}^{p} y^{(p)}, \tag{2.6a}$$

$$D_p^{p-1} = h_p^{p-1} y^{(p-1)} + h_p^p e_1^{(p-1)} - c_{pp} h_p^p y^{(p)}, \tag{2.6b}$$

where functions are evaluated at $H/2$ unless stated otherwise, and terms in $O(H^{p+1})$ have been dropped, as they are in the remainder of this discussion.

Equation (2.5) yields an asymptotically correct value of $H^p y^{(p)}$ which can be substituted into (2.6) to get a pair of equations which can be solved for asymptotically correct values of $H^{p-1}y^{(p-1)}$ and $H^p e_1^{(p-1)}$.

This process can be repeated. At the next step we examine D_m^{p-2} for $m = p - 2$, $p - 1$, and p. We get three equations which can be solved for $H^{p-2}y^{(p-2)}$, $H^{p-1}e_1^{(p-2)}$, and $H^p e_2^{(p-2)}$.

The "cost" of this process in terms of function evaluations can be seen to be $1 + p(p - 1)/2$ because the interval H is integrated by Euler's method using m steps of size H/m for $m = p, p - 1, \dots, 1$. The first of these takes p evaluations of f, but subsequent ones take $p - 2, p - 3, \dots, 0$ because the initial value of y' only has to be calculated once. An example of this is presented in Section 3, where it is shown that this is equivalent to an explicit RK method.

3. EXPLICIT RUNGE-KUTTA METHODS

The extrapolation technique of Section 2 can be viewed as a Runge-Kutta method. We first illustrate this for the case $p = 3$. Let $h = H/3$ and consider an autonomous system. We have

$$
\begin{aligned}
y_1^3 &= y_0 + hf(y_0) = y_0 + k_1, && \text{where} \quad k_1 = hf(y_0),\\
y_2^3 &= y_1^3 + hf(y_1^3) = y_0 + k_1 + k_2, && \text{where} \quad k_2 = hf(y_1^3),\\
y_3^3 &= y_2^3 + hf(y_2^3) = y_0 + k_1 + k_2 + k_3, && \text{where} \quad k_3 = hf(y_2^3),\\
y_1^2 &= y_0 + \tfrac{3}{2}hf(y_0) = y_0 + \tfrac{3}{2}k_1,\\
y_2^2 &= y_1^2 + \tfrac{3}{2}hf(y_1^2) = y_0 + \tfrac{3}{2}k_1 + \tfrac{3}{2}k_4, && \text{where} \quad k_4 = hf(y_1^2),\\
y_1^1 &= y_0 + 3hf(y_0) = y_0 + 3k_0.
\end{aligned}
\tag{3.1}
$$

We use the difference formula given in (3.2) below where all derivatives are evaluated at $t = 3h/2$. $O(h^4)$ terms are neglected.

$$
\begin{aligned}
D_3^3 &= y_3^3 - 3y_2^3 + 3y_1^3 - y_0 = h^3 y^{(3)},\\
D_2^3 &= y_3^3 - y_2^3 - y_1^3 + y_0 = 2h^2 y^{(2)} + 2h^3 e_1^{(2)},\\
D_2^2 &= y_2^2 - 2y_1^2 + y_0 = \left(\frac{3h}{2}\right)^2 y^{(2)} + \left(\frac{3h}{2}\right)^3 e_1^{(2)},\\
D_1^3 &= y_2^3 - y_1^3 = hy^{(1)} + h^2 e_1^{(1)} + h^3 e_2^{(1)} + \frac{h^3}{24} y^{(3)},\\
D_1^2 &= y_2^2 - y_0 = 3hy^{(1)} + \frac{9}{2}h^2 e_1^{(1)} + \frac{27}{4}h^3 e_2^{(1)} + \frac{9}{8}h^3 y^{(3)},\\
D_1^1 &= y_1^1 - y_0 = 3hy^{(1)} + 9h^2 e_1^{(1)} + 27h^3 e_2^{(1)} + \frac{27}{24}h^3 y^{(3)}.
\end{aligned}
\tag{3.2}
$$

From these we can solve for $h^s y^s$ to get

$$
\begin{aligned}
h^3 y^{(3)} &= D_3^3 + O(h^4),\\
h^2 y^{(2)} &= \tfrac{3}{2}D_2^3 - \tfrac{8}{9}D_2^2 + O(h^4),\\
hy^{(1)} &= \tfrac{9}{2}D_1^3 - \tfrac{4}{3}D_1^2 + \tfrac{1}{6}D_1^1 + \tfrac{9}{8}D_3^3 + O(h^4).
\end{aligned}
\tag{3.3}
$$

From (3.2) and (3.1) the D_m^k can be expressed as combinations of the k_i, hence (3.3) is equivalent to an RK method with $q = 4$, the matrix of β coefficients given by

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \frac{3}{2} & 0 & 0 & 0 \end{bmatrix} = \{\beta_{ij}\}$$

and the matrix of Γ coefficients given by

$$\Gamma_{3/2} = \begin{bmatrix} -\frac{3}{8} & -\frac{1}{6} & 1 \\ \frac{9}{4} & 0 & -2 \\ \frac{9}{8} & \frac{3}{2} & 1 \\ -2 & -\frac{4}{3} & 0 \end{bmatrix} = \{\gamma_{js}\}.$$

This gives estimates for the derivatives at $H/2 = 3h/2$. Estimates of the derivatives and/or function values to the same order of accuracy can be obtained at any point within a constant multiple of the interval H. We used the midpoint above to take advantage of symmetry, but for convenience in the discussion below we consider the problem of computing the derivatives at the origin. In that case the matrix above becomes

$$\Gamma_0 = \begin{bmatrix} 1 & -\frac{5}{3} & 1 \\ 0 & 3 & -2 \\ 0 & 0 & 1 \\ 0 & -\frac{4}{3} & 0 \end{bmatrix}.$$

Obviously, the estimate for $hy'(0)$ is simply k_1.

In the example above it took 4 function evaluations for third order. Using the technique of Section 2, it would take 7 function evaluations for fourth order and 11 for fifth order. It is well known that Runge-Kutta methods of third, fourth, and fifth orders are possible with 3, 4, and 6 function evaluations, respectively. This naturally raises the question, "Can the first p derivatives be found to accuracy $O(h^{p+1})$ with fewer than the $1 + p(p-1)/2$ function evaluations used by the technique of Section 2?" (Note that this is not the same problem as finding embedded RK methods of several orders such as the RK Fehlberg method. See Bettis [1].) For $p \leq 3$ the answer is no, but for $p = 4$ it is possible in six function evaluations but no less. The cases $p = 1$ and $p = 2$ are trivial. The problem is examined in detail for $p = 3$ below and for $p = 4$ in the appendix of [8], on which this paper is based.

3.1 Nonexistence of a Three-Stage Method of Third Order

In the following discussion all variables are evaluated at $t = 0$, $y = y_0$, unless specified otherwise. Thus we can write $hy' = hf = k_1$. For a system of m equations in the dependent variables $y^1, y^2, \ldots, y^m$,

$$h^2y'' = h^2 \sum_{i=1}^{m} \frac{\partial f}{\partial y^i} f^i. \tag{3.4}$$

We write the right-hand side as $h^2 f_1 f$. Similarly,

$$h^3 y''' = h^3 \sum_{i,j=1}^{m} \left(\frac{\partial^2 f}{\partial y^i \partial y^j} f^i f^j + \frac{\partial f}{\partial y^i} \frac{\partial f^i}{\partial y^j} f_j \right)$$
$$= h^3 (f_2 f^2 + f_1^2 f) \qquad \text{(definition of } f_2\text{)}. \tag{3.5}$$

Writing $\alpha_i = \sum_{j=1}^{i} \beta_{ij}$, the k_i in (2.1) can be expanded to

$$k_1 = hf,$$
$$k_2 = hf + \alpha_1 h^2 f_1 f + \frac{\alpha_1^2}{2} h^3 f_2 f^2 + O(h^4), \tag{3.6}$$
$$k_3 = hf + \alpha_2 h^2 f_1 f + \frac{\alpha_2^2}{2} h^3 f_2 f^2 + \alpha_1 \beta_{22} h^3 f_1^2 f + O(h^4).$$

Therefore, the second equation of (2.1) can be written as

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \alpha_1 & \alpha_2 \\ 0 & \alpha_1^2/2 & \alpha_2^2/2 \\ 0 & 0 & \alpha_1\beta_{22} \end{bmatrix} \Gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

where Γ is the matrix $[\gamma_{is}]$. The rows of this equation correspond to the terms f, $f_1 f$, $f_2 f^2$, and $f_1^2 f$, respectively. Clearly, the first column of Γ is $[1, 0, 0]^T$ and the first row is such that the sum of all rows is [1, 0, 0]. Thus, the first row and column can be calculated if values can be found to satisfy the remaining six equations in (3.7). Therefore, we drop the first row and column in (3.7) to get

$$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_1^2 & \alpha_2^2 \\ 0 & \alpha_1\beta_{22} \end{bmatrix} \Gamma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

where Γ is now a 2 by 2 matrix. If this is to have a solution, $\alpha_1\beta_{22} \neq 0$. The equation in the (3, 1) position of (3.8) implies $\gamma_{21} = 0$. Then the equation in the (2, 1) position implies $\gamma_{11} = 0$ so that it is impossible to satisfy the equation in the (1, 1) position. Hence, a three-stage third-order method does not exist. However, we have demonstrated the existence of a four-stage method.

3.2 Fourth-Order Methods

Equation (3.4) must be satisfied as identities in each of the elementary differentials of f of orders up to p. This leads to a large system of nonlinear equations for large p. The number of elementary differentials for $p \leq 5$ are given in Table I. The elementary differentials are represented in a prefix operator notation: for example, f_3 is a ternary operator given by

$$f_3\, abc \equiv \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \frac{\partial^3 f}{\partial y_i \partial y_j \partial y_k} a^i b^j c^k,$$

where m is the dimension of the system and a, b, and c are the three vector operands of f_3. For an alternative notation due to Butcher, see Stetter [14, p. 111ff.]. There are 8 elementary differentials of orders up to 4. The second

Table I. Elementary Differentials of f

Order	Number	Representation
1	1	f
2	1	$f_1 f$
3	2	$f_2 f^2, f_1^2 f$
4	4	$f_3 f^3, f_2 f_1 f^2, f_1 f_2 f^2, f_1^3 f$
5	9	$f_4 f^4, f_3 f_1 f^3, f_1 f_3 f^3, f_2^2 f_3, f^2 f_1^2 f^2,$ $f_2(f_1 f)^2, f_1 f_2 f_1 f^2, f_1^2 f_2 f^2, f_1^4 f^4$

equation of (2.1) can be written as

$$A\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the rows correspond to f, $f_1 f$, $f_2 f^2$, $f_1^2 f$, $f_3 f^3$, $f_2 f_1 f^2$, $f_1 f_2 f^2$, and $f_1^3 f$, respectively. For a q-stage method A is an 8 by q matrix whose entries are completely determined by the β_{ij}, and Γ is a q by 4 matrix. As before, the first row and column of all matrices can be discarded because the first row of A is $[1, 1, \dots, 1]$ and the first column is $[1, 0, \dots, 0]^T$. This leaves a system of 21 nonlinear equations in $3(q-1) + q(q-1)/2$ unknowns. Counting unknowns is of no direct value in determining the existence of solutions, although it can be a guide to the prospects. Although for $q = 5$ there are 21 equations in 22 unknowns, it is shown in the appendix to [8] that no solution exists. However, for $q = 6$, when there are 21 equations in 30 unknowns, there exists a 9-parameter family of solutions. This is given in the appendix to [8]. A particular case of this is

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ \frac{3}{4} & 0 & \frac{9}{4} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 2 & 0 & 0 \\ \frac{1}{12} & 2 & \frac{1}{4} & \frac{2}{3} & 2 & 0 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} 1 & -\frac{5}{6} & \frac{4}{9} & -\frac{1}{9} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{4}{9} & \frac{1}{9} \\ 0 & \frac{7}{3} & -\frac{19}{9} & \frac{7}{9} \\ 0 & -3 & \frac{10}{3} & -\frac{4}{3} \\ 0 & 1 & -\frac{11}{9} & \frac{5}{9} \end{bmatrix}.$$

Naturally, one wonders about higher order techniques. For example, since there are 17 elementary differentials of orders up to 5, we are led to a system of 16 by 4 equations (after discarding the first row and column) in $(q + 8)(q - 1)/2$ unknowns. Nine is the first value of q for which the number of unknowns exceeds the number of equations, but it is not known whether a solution exists for $q = 9$; neither does it seem practically important since the increase in the amount of work does not seem to justify the small increase in order.

4. IMPLICIT RUNGE-KUTTA METHODS AND STIFF EQUATIONS

It is well known that explicit methods do not function well for stiff equations. However, immediately after a discontinuity a differential equation is not normally stiff because the solution is in a transient region where the step must be small to control stability. This can be seen by considering the equation

$$y' = \lambda(y - F(t)) + g(t), \tag{4.1}$$

where $g = F'$ almost everywhere and $\lambda \ll 0$. Its solution is

$$y = ce^{\lambda t} + F(t). \tag{4.2}$$

After a short time, the value of the term $ce^{\lambda t}$ is small compared with the value of F, so the behavior of F dominates the solution. However, if there is a discontinuity in the right-hand side of (4.1) at some $\hat{t}$, and this discontinuity is not consistent with $g = F'$, then the solution (4.2) will have a discontinuity in c and F, making the first term significant for another short time. If this is the case, the explicit methods discussed earlier should be used. However, in some cases the discontinuity may not stimulate the rapidly decaying components sufficiently so that a step size large compared with $1/\lambda$ can be used in principle. If the explicit methods of the previous sections are applied to (4.1), we obtain approximations to the scaled derivatives of F plus polynomials in $h\lambda$. For example, if the six-stage method of order 4 is used, the approximation to the pth derivative of y will be

$$h^p y^{(p)} = h^p F^{(p)} + c(h\lambda)^p + O(h^5) + cO(h^5\lambda^5) + O(h^5\lambda^4),$$

where the $O(h^5)$ term depends on derivatives of F, but not on c and λ; the $O(h^5\lambda^5)$ is independent of the derivatives of F, but the $O(h^5\lambda^4)$ term has coefficients which depend on the derivatives of F. If c is small, we would like the pth derivatives of F to dominate this expression. However, if $h\lambda$ is large, the $O(h^5\lambda^4)$ term may well dominate. For these problems it is worth considering implicit RK methods. It is known (Butcher [2]) that q-stage implicit RK methods can achieve order $2q$. It is also known (Ehle [4]) that such methods are A-stable. In fact, we are not concerned with A-stability or its variants such as stiff stability because only a single RK step is taken, while A-stability is concerned with the limiting behavior as the number of steps becomes infinite. What we do need are approximations to the first p scaled derivatives that are not contaminated with errors due to large Jacobian elements when the corresponding components of the solution are absent, but which give some indication when the components are present because in the latter case we must reduce the step size to handle the transients. Thus, if we are allowed a local error of ϵ, we can ignore components whose size is smaller than ϵ, provided they do not contaminate the solution. However, components of size larger than ϵ must be integrated correctly. We follow the usual practice of doing a linear analysis for the test equation $y' = \lambda y$ and ask that the approximations to the scaled derivatives be bounded for all λ in the negative half-plane. Suppose that the approximation to the pth scaled derivative is bounded by δ_p when the method is applied to the test equation with $y(0) = 1$. Then a scaled derivative larger than $\epsilon\delta_p$ can be used as an indication that there is a significant component which must be integrated correctly by step-size reduction.

When a q-stage implicit RK method is applied to the test equation, we calculate terms of the form

$$R(h\lambda) = \frac{P(h\lambda)}{Q(h\lambda)} y, \tag{4.3}$$

where

$$Q(\mu) = \det(I - \mu B) = 1 + a_1\mu + \cdots + a_q\mu^q$$

is a polynomial of degree q in μ whose coefficients are completely determined by the matrix B of coefficients β_{ij}, and $P(\mu)$ is a polynomial of degree no greater than q. (See Gear [7].) The scaled derivatives calculated using an implicit form of (2.1) take a similar form: the ith scaled derivative will be

$$\frac{P_i(h\lambda)}{Q(h\lambda)}.$$

Note that the denominator is determined by B and so is independent of i. For consistency with pth order, the pth scaled derivative must take the asymptotic form $(h\lambda)^p + O(h^{p+1})$, from which we see that

$$P_p(\mu) = \mu^p + O(\mu^{p+1}).$$

Hence we conclude that the number of stages q must be at least p. We demonstrate below that the desired properties can be achieved with $q = p$. Since the approximation to the ith scaled derivative must also have order p, we see that

$$\mu^i + O(\mu^{p+1}) = \frac{P_i(\mu)}{1 + a_1\mu + \cdots + a_q\mu^q}.$$

Since P_i is a polynomial, it follows that

$$P_i(\mu) = \mu^i + a_1\mu^{i+1} + \cdots + a_{p-i}\mu^p + O(\mu^{p+1}). \tag{4.4}$$

If $q = p$, the P_i are completely determined by the B matrix, and as $h\lambda = \mu$ approaches infinity, the approximations to the ith scaled derivatives approach a_{p-i}/a_p where $a_0 = 1$. In particular, the estimate for the pth scaled derivative approaches $1/a_p$. This is the value of δ_p which we would like to be of moderate size. Note also that the rational approximations $P_i(\mu)/Q(\mu)$ are bounded for all μ in the negative half-plane if the zeros of $Q(\mu)$ are in the right half-plane and a_p is nonzero.

The existence of p-stage, pth-order implicit methods can be demonstrated without reference to elementary differentials. Assume for the moment that we have the exact values g_i of the $hf(y(t_0 + h\alpha_i))$. If the α_i are all different, the interpolation coefficients β_{ij} can be chosen so that

$$y(t_0) + \sum_{j=1}^{p} \beta_{ij} g_j = y(t_0 + h\alpha_i) + O(h^{p+1}). \tag{4.5}$$

Hence

$$hf(y(t_0) + \sum_{j=1}^{p} \beta_{ij} g_j) = g_i + O(h^{p+2}).$$

Therefore, a solution of

$$k_i = hf(y(t_0) + \sum_{j=1}^{p} \beta_{ij} k_j) \tag{4.6}$$

gives an $O(h^{p+2})$ accurate approximation to g_i. With these p approximations to hy' at p different points, we can obtain approximations to $h^s y^{(s)}$ at t_0 with error no greater than $O(h^{p+1})$. These determine the γ_{is} coefficients uniquely.

It remains to show that the root condition for Q is satisfied. Since the β_{ij} coefficients were completely determined by the α_i in the process above, and these in turn determine the Q polynomial, we can consider the zeros of Q to be functions of the α_i. The existence of α coefficients for which the root condition is satisfied can be demonstrated by considering the Butcher implicit RK methods. The α are the Gaussian quadrature points. For this it is known that Q is a degree p polynomial, all of whose roots are in the right half-plane because the method is A-stable (see Ehle [4]). Thus we see that by using the β_{ij} corresponding to the Butcher methods, the γ_{js} can be chosen to achieve pth-order accuracy. We have also shown that no fewer than p stages can be used.

The cost of these implicit RK methods is high because a set of p nonlinear systems of equations must be solved simultaneously, so it is questionable whether such methods have any application. For this reason, plus the fact that most of the time the problem is not stiff in the starting region, we concentrate on explicit methods.

5. ERROR ESTIMATES AND STEP SELECTION

A useful algorithm must include a technique to estimate the error and to adjust the step size. In the RK starting technique there are two step sizes to be chosen, the step size for the RK process and the first step size for the multistep method. The computational process we propose is as follows. Details are given later.

(1) "Guess" a step size h_R for the RK process based on knowledge of y_0 and y_0'.
(2) Execute the RK starting process using h_R to compute the k_i.
(3) Estimate the effect of truncation and roundoff errors in the RK step from the computed k_i. If these errors are too large, adjust h_R and repeat step (2).
(4) Estimate the step size h_M for the first multistep step from the k_i.

We discuss this process for the fourth-order explicit method proposed in Section 3. The ideas extend to other methods without difficulty.

In the numerical solution of ordinary differential equations all error estimation and step-size control techniques are based on some model of the problem. This may be an asymptotic model in which the first nonvanishing term in a Taylor series for the error is estimated and all higher order terms are neglected, but in many cases this is *not* the model used. Consider, for example, the RK Fehlberg methods [5] in which two approximations of different orders are computed, say $y_A = y + O(h^4)$ and $y_B = y + O(h^5)$. This is a (3–4) method. The difference between y_A and y_B gives an asymptotically correct estimate of the $O(h^4)$ error in y_A, but in most codes the more accurate y_B is used as the result, so the underlying model uses the assumption that the error in y_B is related to that in y_A. Multistep methods in which the corrector order exceeds the predictor order by 1 use a similar assumption because the predictor-corrector difference is an estimate of derivatives of order one lower than those that appear in the corrector error. We are going to use a similar model; namely, we assume that by controlling a lower order derivative we can control the higher order derivatives that appear in the error terms. (One could perform extra computational steps to estimate the error terms asymptotically exactly, but then one would correct the answers to remove those errors, or, as V. Kahan of the University of California at Berkeley has put it succinctly, "One should compute an estimate of the *uncertainty*, not the error.")

With this model, we assume that the error in the multistep method is related to

$$C_M h_M^4 \| y^{(4)} \|,$$

where C_M is the error coefficient of the method used and h_M is the step size to be used. This is to be controlled to be less than the desired tolerance ϵ so if we know C_M and $y^{(4)}$, we choose h_M so that

$$h_M = r\left[\frac{\epsilon}{C_M \| y^{(4)} \|}\right]^{1/4}, \tag{5.1}$$

where $r < 1$ is a "safety factor" which sets h_M smaller than indicated by the model.

Although we compute an $O(h)$ accurate estimate of $y^{(4)}$ in the RK process, we cannot use it directly in (5.1) because it is not uncommon for some initial derivatives to be zero. This would lead to an infinite step h_M. To overcome this problem, we add further assumptions to our model, namely, that the sth derivative $y^{(s)}$ approximately satisfies the relationship

$$\| y^{(s)} \| = \lambda^s \| \hat{y} \|, \qquad s = 1, \ldots, 4 \tag{5.2}$$

for some λ, where $\| \hat{y} \|$ is an estimate of the norm of y to be discussed later. Therefore, we replace the fourth derivative norm $\| y^{(4)} \|$ used in (5.1) with $\| \hat{y}^{(4)} \|$ given by

$$\begin{aligned} \| \hat{y}^{(4)} \| &= \| \hat{y} \| \left[\frac{1}{4} \sum_{s=1}^{4} \left[\frac{\| y^{(s)} \|}{\| \hat{y} \|} \right]^{1/s} \right]^4 \\ &= \lambda^4 \| \hat{y} \| . \end{aligned} \tag{5.3}$$

This can be computed from the output of the RK step.

Now we come to a study of the RK step errors and the determination of h_R. In fact, we are not concerned about the actual errors, only about their influence on the subsequent multistep method. Our model assumes that the truncation error in the computed value of $h_R^s y^{(s)}$ can be estimated by $C_{sR} h_R^4 \| \hat{y}^{(4)} \|$, $s = 2$, 3, and 4. Calculation of the $h_R^s y^{(s)}$ also involves truncation errors when the k_i terms are differenced. The error will have a bound of the form $G_s \| h_R y' \| u$, *where*

$$G_s = \sum_{j=1}^{6} | \gamma_{js} |$$

and u is the roundoff error in computing $f(y)$. When the $h_R^s y^{(s)}$ terms are used in the multistep method (directly if a Nordsieck vector is used, or indirectly to compute y_s, $s = 1, \ldots, k$, if y values are used), the errors introduced by the RK process are multiplied by powers of h_M/h_R to change the step size and by coefficients of the multistep method. We would like to keep the errors introduced by the RK process somewhat smaller than the permitted error tolerance in the multistep method. Therefore, we must require that the truncation and rounding errors in $h_M^s y^{(s)}$ are bounded by $g_s \epsilon$ where the g_s coefficients are determined by the coefficients of the multistep method and the fraction of the error tolerance ϵ that the RK process is permitted to generate in subsequent multistep values. Thus we require that

$$C_{sR} h_R^4 \| \hat{y}^{(4)} \| \left[\frac{h_M}{h_R} \right]^s \leq g_s \epsilon \tag{5.4}$$

and

$$G_s \| h_R y' \| u \left[\frac{h_M}{h_R} \right]^s \leq g_s \epsilon \tag{5.5}$$

for the truncation and roundoff error estimates, respectively. If we use the model form for $\| y^{(s)} \|$ given by (5.2) and (5.3) and use the value of h_M given by (5.1), the truncation error restriction (5.4) becomes

$$C_{sR} (h_R^4 \lambda^4 \| \hat{y} \|)^{(4-s)/4} \left[\frac{r}{C_M^{1/4}} \right]^s \leq g_s \epsilon^{(4-s)/4}. \tag{5.6}$$

For $s = 4$ this gives

$$r \leq \left[\frac{C_M g_4}{C_{4R}} \right]^{1/4}. \tag{5.7}$$

This means that the safety factor r must be restricted. For $s = 2$ and 3, (5.6) gives

$$h_R^4 \lambda^4 \| \hat{y} \| \leq \epsilon \left[\frac{g_s}{C_{sR}} \left[\frac{C_M^{1/4}}{r} \right]^s \right]^{4/(4-s)}. \tag{5.8}$$

The left-hand side can be calculated in the RK step using (5.3). If (5.8) is not satisfied, h_R must be reduced to reduce the RK truncation error. Note that the program calculates λ^4 by (5.3) and makes the simple test

$$h_R^4 \lambda^4 \leq K_U \frac{\epsilon}{\| \hat{y} \|} \tag{5.9}$$

where

$$K_U = \min_{s=2,3} \left[\frac{g_s}{C_{sR}} \left[\frac{C_M^{1/4}}{r} \right]^s \right]^{4/(4-s)}.$$

K_U can be computed from the method coefficients once and for all. We get the roundoff error restrictions by making substitutions from (5.1) to (5.3) into (5.5) to find

$$(h_R \lambda)^{1-s} u G_s \left[\frac{r}{C_M^{1/4}} \right]^s \left[\frac{\| \hat{y} \|}{\epsilon} \right]^{(4-s)/4} \leq g_s, \tag{5.10}$$

which, for $s = 2$, 3, and 4, gives

$$h_R^4 \lambda^4 \geq \left[\frac{u}{C_M} \frac{u\|\hat{y}\|}{\epsilon}\right]^2 \left[\frac{r^2 G_2}{g_2}\right], \tag{5.11}$$

$$h_R^4 \lambda^4 \geq \left[\frac{u}{C_M}\left[\frac{u\|\hat{y}\|}{\epsilon}\right]^{1/3}\right]^{3/2} \left[\frac{r^3 G_3}{g_3}\right]^2, \tag{5.12}$$

and

$$h_R^4 \lambda^4 \geq \left[\frac{u}{C_M}\right]^{4/3} \left[\frac{r^4 G_4}{g_4}\right]^{4/3}. \tag{5.13}$$

Each of the last three inequalities bound h_R from below; that is, if h_R is too small, the roundoff error is too large and h_R must be increased. We are going to argue that the last of these three inequalities is the dominant one, although a code could check all three cases with very little extra effort. If we examine the first parenthesized term on the right-hand side of (5.11) and (5.12) we see that they contain the term $u\|\hat{y}\|/\epsilon$. Since $\epsilon/\|\hat{y}\|$ is the relative error tolerance, we can reasonably demand that this be greater than the relative roundoff u in computing f. Hence we conclude that the first terms on the right-hand sides of (5.11) and (5.12) are no larger than $(u/C_M)^2$ and $(u/C_M)^{3/2}$, while the first term of the right-hand side of (5.13) is $(u/C_M)^{4/3}$. The latter is the largest by order of $(u/C_M)^{-1/6}$. The second terms in the right-hand sides of (5.11)-(5.13) are all of order 1; so we conclude that (5.13) is the most stringent condition. By setting

$$K_L = \left[\frac{u r^4 G_4}{C_M g_4}\right]^{4/3}, \tag{5.14}$$

which can be calculated from the method coefficients, we have only to test to see whether

$$h_R^4 \lambda^4 \geq K_L. \tag{5.15}$$

If it is not, h_R must be increased to reduce roundoff errors.

Finally we come to the problem of "guessing" the first value of h_R for the RK step. From the model assumed in (5.2) we calculate a first approximation to λ based on

$$\lambda = \frac{\|y'\|}{\|\hat{y}\|}. \tag{5.16}$$

This can be done before h_R is known. Now, on the basis of this estimate, we evaluate an upper bound for h_R from (5.9) and a lower bound from (5.15). The initial h_R used is the geometric mean of these two bounds. If, after the RK step, one of the bounds is violated based on the new estimate of λ, the geometric mean of the new bounds is used to redo the RK step.

There are several programming considerations, that is to say, heuristic solutions to awkward difficulties. If the initial estimate of λ from (5.16) is too small, h_R will be too large. This problem is avoided by limiting h_R to a maximum of 1. Because different components in a system may be scaled very differently, the norms used are scaled component by component so that errors can be relative in each component. Normally the scaling is proportional to values of the corresponding components of y with elements smaller than some tolerance being replaced by that tolerance. L_2 norms are used after this scaling. After this scaling, $\|\hat{y}\|$ is taken to be 1. Finally, we might find that $K_U\epsilon < K_L$, in which case it is not possible to satisfy (5.9) and (5.15) simultaneously. This can be taken as an indication that ϵ is too small. (Note from (5.14) that $K_L = O(u^{4/3})$, implying a request for an error tolerance ϵ smaller than the roundoff.) In this case, it seems reasonable to reject the error request as too small.

6. NUMERICAL IMPLEMENTATION AND TESTS

A code has been appended to Hindmarsh's integrator [9] to implement this algorithm. The fourth-order starter is used to provide up to fourth derivatives so that the multistep method can start with an order-4 method. The coefficient values used are

$C_M = \frac{1}{24}$ for Adams, $\frac{1}{4}$ for BDF;

$C_{sR} = \frac{1}{24}$, $s = 2$, 3, and 4;

$g_s = \frac{1}{4}$, $s = 2$, 3, and 4;

$u = 8$ unit rounding errors in the machine used;

$r = \frac{1}{2}$.

This value of r does not violate (5.7).

These values lead to

$K_L = 718\delta^{4/3}$ Adams,

$K_L = 40.3\delta^{4/3}$ BDF,

where δ is the unit rounding error in the computer used, and

$K_U = 24$ Adams, 144 BDF.

Admittedly, the choice of these coefficients is open to question; their only defense is an appeal to the model of errors and selection of safety factors. The tests that follow indicate that they are a reasonable selection for at least some problems and tolerances.

A nonlinear test equation was constructed for some of the tests as follows:

Let $\mathbf{y}$ and $\mathbf{u}$ be s-element vectors, and let $\mathbf{g}(t)$ and $\mathbf{p}(\mathbf{u})$ be s-element vector functions, the former being a function of the scalar time variable t, and the latter being an invertible, componentwise function of the vector argument $\mathbf{u}$. (For example, $\mathbf{p}(\mathbf{u})$ could be the exponential.) Let A and B be any matrices; A should be nonsingular. Define u by the equation

$$\mathbf{u}' = A[B[A^{-1}\mathbf{u} - \mathbf{g}(t)] + \mathbf{g}'(t)]$$

and define $\mathbf{y}$ by

$$\mathbf{y} = \mathbf{p}(\mathbf{u}).$$

Hence the differential equation for $\mathbf{y}$ is

$$\mathbf{y}' = \mathbf{p}'\mathbf{u}'.$$

Table II. Effect of Using Starter in Hindmarsh Code

EPS	fn evals Hindmarsh	fn evals modified	h in RK	h for Adams start
0 1D−02	6	9	0 13523D−02	0.44751D−01
0.1D−03	9	10	0.10141D−02	0.25163D−01
0.1D−04	11	13	0 76047D−03	0.14149D−01
0 1D−05	16	14	0.57027D−03	0.79558D−02
0 1D−06	20	17	0 42764D−03	0.44736D−02
0.1D−07	26	20	0.32069D−03	0.25156D−02
0 1D−08	32	23	0.24048D−03	0.14146D−02
0.1D−09	40	28	0.18034D−03	0.79545D−03
0 1D−10	50	36	0.13523D−03	0.44731D−03
0.1D−11	62	43	0 10141D−03	0.25154D−03

The matrix B can be chosen to make the problem stiff or not, while the function $\mathbf{g}$ can be chosen to get a variety of components in the solution. The matrix A serves to "mix up" the components, and the nonlinear function $\mathbf{p}$ serves to bring in nonlinearities.

The data in Table II indicate the effect of the starter on the Hindmarsh code when the problem described above was integrated over the interval (0.0, 0.1) using the Adams methods. The particular problem tried had three components, $p(u) = \exp(u)$, the matrices A and B were given by

$$A = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -4 & 3 \\ 0 & -3 & -4 \end{bmatrix},$$

and the vector $\mathbf{g}$ was $[\sin(t), t, (1-t)/(1+t) -1]$. The initial values were $y_i = 1$ for $i = 1, 2,$ and 3.

The errors in the two versions were not significantly different. For some values of the error tolerance EPS the original version was slightly more accurate; in other cases the modified version was better. (Both versions gave errors approximately equal to EPS in this problem.) The second and third columns of Table II indicate the number of function evaluations used by the original Hindmarsh code and the modified form for a range of error tolerance parameters EPS. From this it can be seen that the modification is beneficial only for small tolerances: it actually uses more function evaluations for large tolerances. This is plausible since fourth order is probably too high for the larger tolerances shown. However, even in those cases, the loss is very little and the actual time lost is less because six of the function evaluations are used in the RK starter with very little overhead compared to the high overhead in the automatic multistep method.

A number of tests were run to determine whether the step-size-selection scheme was appropriate. A set of very simple equations, including a mildly stiff problem with stiffness ratio of about 100, were used. For each problem, the following procedure was followed. For each EPS in the range 10^{-m}, $m = 1, 2, \ldots, 11$, the starter was executed with the RK step size forced to be 2^{-k}, $k = 1, 2, \ldots, 15$, in turn. An error was calculated by forming the difference between the true solution at a point a distance h_M from the origin and the value that would be predicted using the starting values at the same point. Here h_M is the step size estimated by the starter for the first Adams step. The RK step size that led to the smallest such error was selected for each EPS. This would be the most desirable one to have chosen. Then the RK starter was allowed to choose its own RK step size using the procedure given in Section 5. For nonstiff problems, the RK step size chosen automatically was within a factor of 2 of the optimum for EPS $< 10^{-3}$. However, as was pointed out in Section 5, the actual step size used for the RK starter is not that critical; what is important is that a good step size be chosen for the first application of the multistep method and that the error due to the RK method be below the tolerance. The step size selected for the multistep method is almost independent of the RK step size because it depends only on the estimate made for λ which is determined by the size of the calculated derivatives, not the errors in them. In all nonstiff cases, the error in the predicted value was well below the tolerance EPS. In the mildly stiff case, the recommended RK step size tended to be too large by a factor of about 4, and this led to an error greater than EPS when EPS was less than 10^{-5}. The worst case was an error 30 times too large when EPS was 10^{-11}.

It should be noted that one of the test problems was a problem that would be stiff after its initial transient died out, but was such that a small step size would be needed for accuracy initially. This gave results compatible with the nonstiff results.

These limited tests indicate that the method may have value for problems which do not start in a stiff region, as is true for most problems. An additional benefit is that the scheme appears to give a reasonable estimate of the starting step size, neither too large nor too small. Detailed examination of the step sizes used in the original and modified Hindmarsh codes indicated that the bulk of additional function evaluations were being used to arrive at a reasonable step size, not to maintain the requested error, while in the modified version, the initial step size selected by the starter seldom caused the error test to fail and was within a small factor of the step size ultimately chosen by either method at fourth order. It may be that an improvement in the initial step-size-selection algorithm used in a code would be as effective as the proposed starter—or it may mean that the major benefit in this starter is its ability to select the step size.

ACKNOWLEDGMENT

I would like to thank F. Cellier of ETH Zurich for discussions which led to the study of this problem.

REFERENCES

1. BETTIS, D.G. Efficient embedded Runge-Kutta methods In *Numerical Treatment of Differential Equations*, vol. 631, R. Burlirsch, R.D. Grigorieff, and J. Schroder, Eds., Lecture Notes in Mathematics, Springer-Verlag, New York, 1976.
2. BUTCHER, J.C. Implicit Runge-Kutta processes *Math. Comput. 18* (1964), 50-64.
3. CARVER, M.B. Efficient handling of discontinuities and time delays in ordinary differential equations. In Proc Simulation '77 Symp., Montreux, Switzerland, M. Hamza, Ed., Acta Press, Zurich, Switzerland
4. EHLE, B.L. *A*-stable methods and Pade approximations to the exponential. *SIAM J. Math. Anal. 4* (1973), 671-680
5. FEHLBERG, E. Klassiche Runge-Kutta Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und inre Anwendung auf Warmeleitungsprobleme. *Computing 6* (1970), 61-71.

6. GEAR, C W. DIFSUB for the solutıon of ordinary dıfferentıal equations. *Commun. ACM 14*, 3 (March 1971), 185-190.
7. GEAR, C W. Ratıonal approximatıons by implıcıt Runge-Kutta schemes. *BIT 10* (1970), 20-22.
8. GEAR C.W Runge-Kutta starters for multıstep methods. Rep. 78-938, Dept. Comput. Scı., Univ. Illinois, Urbana, 1978.
9. HINDMARSH A.C. GEAR: Ordinary differential equatıon solver Rep. UCID-30001, Rev. 3, Lawrence Livermore Lab., Lıvermore, Calıf., 1974.
10. HULL, T.E., ENRIGHT, W.H., FELLEN, B.M., AND SEDGWICK, A.E. Comparing numerıcal methods for ordinary differentıal equations. *SIAM J. Numer. Anal. 9* (1972), 603-637.
11. KROGH, F T VODQ/SVDQ/DVDQ—Varıable order integrators for the numerical solution of ordinary dıfferentıal equatıons. Sec. 314, Subroutıne Wrıte-up, Jet Propulsıon Lab., Pasadena, Calıf., 1969
12. NORDSIECK, A. On the numerical ıntegratıon of ordinary differential equations. *Math. Comput. 16* (1962), 22-49.
13 SHAMPINE, L.F., AND GORDON, M.K. *Computer Solutıon of Ordinary Differential Equatıons: The Inıtıal Value Problem.* Freeman, San Francısco, 1975.
14 STETTER, H.J. *Analysıs of Dıscretızatıon Methods for Ordınary Dıfferentıal Equations*, vol 23, Sprınger Tracts in Natural Philosophy, Sprınger-Verlag, New York, 1973.
15. STETTER, H.J Asymptotıc expansıons for the error of dıscretizatıon algorithms for nonlınear functional equations. *Numer. Math. 7* (1965), 18-31

Received August 1978; revised October 1979, accepted March 1980