MATHEMATICS OF COMPUTATION
VOLUME 39, NUMBER 160
OCTOBER 1982, PAGES 491-509

Two Classes of Internally S-Stable Generalized Runge-Kutta Processes Which Remain Consistent With an Inaccurate Jacobian

By J. D. Day and D. N. P. Murthy

Abstract. Generalized Runge-Kutta Processes for stiff systems of ordinary differential equations usually require an accurate evaluation of a Jacobian at every step. However, it is possible to derive processes which are Internally S-stable when an accurate Jacobian is used but still remain consistent and highly stable if an approximate Jacobian is used. It is shown that these processes require at least as many function evaluations as an explicit Runge-Kutta process of the same order, and second and third order processes are developed. A second class of Generalized Runge-Kutta is introduced which requires that the Jacobian be evaluated accurately less than once every step. A third order process of this class is developed, and all three methods contain an error estimator similar to those of Fehlberg or England.

1. Introduction. In this paper we are concerned with the approximate numerical integration of nth order stiff systems [22, p. 228] of ordinary differential equations of the form

$$y' = f(y); \qquad y(x_0) = y. \tag{1.1}$$

The most widely used algorithms are those based on multistep formulas (e.g., [18]). These methods are efficient, especially for computing accurate solutions (i.e., when the specified error tolerance is small). However there is a limit to the level of numerical stability that a multistep method can possess. In particular, no multistep method of order greater than 2 can be A-stable [22, p. 233], and consequently multistep methods usually satisfy a relaxed stability criterion, such as $A(\alpha)$-stability or Stiff-stability [22, p. 233]. Thus the methods are not suited to problems in which the eigenvalues of the Jacobian contain large imaginary parts. In addition, the stability analysis is linear, and numerical experience indicates that, for very stiff nonlinear systems, A-stable methods are inadequate. This led Prothero and Robinson [25] to define S-stability, which, although it also uses a linear scalar test equation, was claimed by the authors to be suitable for nonlinear systems, in which the eigenvalues are widely separated. The criterion has been applied to implicit Runge-Kutta methods which were originally defined by Butcher [7]. These S-stable implicit Runge-Kuttas are thus more reliable, in the sense that a wider range of stiff systems can be computed with them, but less efficient than the multistep methods, in the sense that more computation per step is required. In attempting to reduce the

Received May 27, 1980; revised January 22, 1981 and October 15, 1981.
1980 *Mathematics Subject Classification.* Primary 65L05.
Key words and phrases. Generalized Runge-Kutta procedure, semi-implicit Runge-Kutta procedure, approximate Jacobian, stiff differential equations, L-stability, A-stability, S-stability, Internal S-stability.

©1982 American Mathematical Society
0025-5718/82/0000-0318/$05.00

computation per step of the implicit Runge-Kuttas, we consider generalized Runge-Kuttas which are linearly implicit, and hence eliminate the need for solving nonlinear algebraic equations, which must be solved by an iterative technique such as Newton-Raphson, and hence require additional function evaluations for every iteration at every step. It should be noted, however, that these methods may require less than one Jacobian evaluation and matrix factorization per step, or several Jacobian evaluations and matrix factorizations per step, depending on the convergence of the Newton-Raphson iterations. Following Verwer [29], we describe a *Generalized Runge-Kutta* process for solving Eq. (1.1) by

$$y_{n+1} = y_n + \sum_{i=1}^{v} A_{(v+1)i} k_i, \tag{1.2}$$

where

$$k_1 = h_n f(y_n), \qquad k_i = h_n f\left(y_n + \sum_{j=1}^{i-1} A_{ij} k_j \right), \qquad i = 2, 3, \ldots, v,$$

and where A_{ij}, $i = 2, \ldots, v+1$; $j = 1, \ldots, i-1$, are nth order square matrices which are functions of the Jacobian matrix $\partial f(y(x))/\partial y|_{x=x_n}$ $(\equiv J_n)$ or some approximation to the Jacobian. We assume that the system (1.1) is *inherently stable* (i.e., the eigenvalues $(\lambda_j, j = 1, \ldots, n)$ of the Jacobian are such that $\mathrm{Re}(\lambda_j) < 0$, $j = 1, \ldots, n$). It is possible to define several classes of Generalized Runge-Kutta [10] depending on the type of matrix function used for the A_{ij}. For example, if we use a rational polynomial function such as

$$A_{ij} = a_{ij} \left[I + \sum_{q=1}^{r} \alpha_q h_n^q J_n^q \right] \left[I + \sum_{q=1}^{r+l} h_n^q b_k J_n^q \right]^{-1}, \tag{1.3}$$

where α_q, $q = 1, \ldots, r$, and b_q, $q = 1, \ldots, r+l$, and a_{ij} are real scalars, and $l \geqslant 1$ for stability reasons (see Theorem 1), then at every step one or more matrices must be factorized and one or more systems of linear algebraic equations must be solved with this factorized matrix. In the interests of efficiency, we will use a matrix of the form

$$[I - h_n b J_n]^{-(r+l)} \tag{1.4}$$

since a matrix $*$ matrix product J_n^2 requires n^3 operations, which is approximately three times the work required to factorize the matrix. Since the functions A_{ij} all premultiply some vector, the powers of J_n in the numerator are not matrix $*$ matrix multiplications, but rather matrix $*$ vector multiplications. The use of (1.4) means that only one matrix factorization per step is necessary, although there will be several matrix $*$ vector multiplications and several solutions of linear systems of algebraic equations. If the Jacobian is expensive to compute, it is interesting then to consider a class of methods which uses an approximation to the Jacobian $(\bar{J}_n)$, so that we have typically

$$A_{ij} = a_{ij} \left[I + \sum_{q=1}^{r} \alpha_k h_n^q \bar{J}_n^q \right] \left[I - h_n b \bar{J}_n \right]^{-(r+l)}. \tag{1.5}$$

As we will see in Section 3, the form (1.5) results in a requirement for more function evaluations per step than the form (1.3) and (1.4). To reduce this requirement, and

also the requirement that a Jacobian be evaluated at every step, we may use a Jacobian which has been evaluated accurately at a previous step, but not reevaluated for the current step. Such a function might be of the form

$$A_{ij} = a_{ij}\left[I + \sum_{q=1}^{r} \alpha_q h_n^q J_{n-k}^q\right][I - h_n b J_{n-k}]^{-(r+l)}, \tag{1.6}$$

where J_{n-k} is the Jacobian evaluated at x_{n-k}, where $x_n - x_{n-k} = \Sigma_{j=1}^{k} h_{n-j}$. Explicit methods of order p, with v stages, are designated (p, v). The number of function evaluations per step is v, and this gives an indication of the efficiency of a particular method. For the Generalized Runge-Kuttas, we also have the operations of solving linear algebraic systems (usually by back substitution [17, p. 51]) and multiplication of a vector by a matrix. Thus it is appropriate to describe them by (p, v, j, r), where j is the number of back substitutions per step, and r the number of matrix $*$ vector multiplications. In this paper, we designate the form (1.5) as Type 1, and the form (1.6) as Type 2.

Many linearly implicit Runge-Kutta processes (sometimes called Rosenbrock methods) have been developed. Some require more than one Jacobian evaluation per step. However, the others are processes of Type (1.3) (see for example [1]–[5], [8]–[11], [19], [20], [23], [24], [26]). Little work has been done on Type 1 processes. Eitelberg [12] and Steihaug and Wolfbrandt [27] have developed second order processes. In this paper we develop second and third order processes of this type, which we consider to be superior either in terms of efficiency (less computation required per step), accuracy, or reliability (more stable). We are unaware of any work which has been done on Type 2 processes, and we will develop a third order formula. We will construct formulas of both classes which are Internally S- (or $S(\alpha)$-) stable [29] when the Jacobian is accurately evaluated. When the Jacobian is not evaluated accurately at each step, the stability of the processes is uncertain, and we expect a deterioration in stability as the difference between J_n and either $\bar{J}_n$ or J_{n-k} increases. The analysis of this problem has not been considered in this paper. However, we expect the onset of instability to be detected by the local truncation error estimator which will force the step size to become smaller. We will use an imbedding type error estimator [21, p. 164] which is easy to obtain from the Generalized Runge-Kutta methods of the type described in this paper, and requires less computation per step than the well-known Richardson Extrapolation, or method of interval halving [22, p. 130]. The processes were tested on two systems taken from the literature.

2. Stability. Since we intend to construct formulas which are Internally S-stable, when an exact Jacobian is used, we will summarize some stability results for methods of the Type (1.3). (The results are applicable to Type 1 and Type 2 formulas if an accurately evaluated Jacobian is used.) If an integration method applied to the scalar test equation

$$y' = \lambda y \tag{2.1}$$

yields the function $R(Z)$, such that

$$y_{n+1} = R(Z) y_n, \tag{2.2}$$

where the complex number $Z = h_n\lambda$, and $x_{n+1} = x_n + h_n$, then we have the definitions:

Definition 1 [29]. A rational polynomial $R(Z)$ is said to be (a) *A-acceptable* if $|R(Z)| < 1$ whenever $\mathrm{Re}(Z) < 0$; (b) *Strongly A-acceptable*, if it is A-acceptable and satisfies $\lim |R(Z)| < 1$ as $\mathrm{Re}(Z) \to -\infty$; (c) *L-acceptable* if it is A-acceptable and satisfies $\lim R(Z) = 0$ as $\mathrm{Re}(Z) \to -\infty$. □

Definition 2 [29]. Let $\bar{h}$ be any positive real number. Then the integration method is said to be *S-stable* if, for a scalar differential equation of the form

$$y' = g'(x) + \lambda(y - g(x)), \tag{2.3}$$

ξ_n $(= g(x_n) - y_n)$ is uniformly bounded with n for all λ with $\mathrm{Re}(\lambda) < 0$ and all $h_n \in (0, \bar{h})$. □

It should be noted in the above definition that the exact solution of (2.3) at x_n is $g(x_n)$ and the computed solution is y_n. Verwer [29] analyzed the S-stability of Generalized Runge-Kutta methods, and following Verwer we characterize a Generalized Runge-Kutta by the array

(2.4)

$$A_m^* \equiv \left[\begin{array}{c} \Lambda \\ \hline \Lambda_m^T \end{array}\right] \left[\begin{array}{cccccc} 0 & & & & & \\ A_{21}(Z) & & & & & \\ A_{31}(Z) & A_{32}(Z) & & & & \\ \vdots & & & & & \\ A_{m_1}(Z) & & \cdots & A_{m(m-1)}(Z) & 0 & \\ \hline A_{(m+1)1}(Z) & A_{(m+1)2}(Z) & \cdots & A_{(m+1)}(Z)_{(m-1)} & A_{(m+1)m}(Z) & \end{array}\right]$$

for the mth stage, where $m \in [1, v]$. The parameters A_{ij}, $i = 2,\ldots,m+1$; $j = 1,\ldots,i-1$, are scalars, and are functions of the complex number Z $(\equiv h_n\lambda)$. The functions $A_{ij}(Z)$ are the functions $A_{ij}(h_nJ_n)$ with h_nJ_n replaced by Z to give a scalar function rather than a matrix function. Verwer [29] introduced the stability function for the mth stage

$$R^{(m)}(Z) = 1 + Z\Lambda_m^T[I - Z\Lambda(Z)]^{-1}e, \qquad m = 1,2,\ldots,v, \tag{2.5}$$

where $e^T = [1, 1,\ldots, 1]$, the mth order unit vector, and proved the following theorem about methods with an accurately evaluated Jacobian.

THEOREM 1 [29]. *A Generalized Runge-Kutta method is S-stable if*

(a) $R^{(v)}(Z)$ *is strongly A-acceptable, and*

(b) *its coefficient functions* A_{ij}, $i = 2,\ldots,v+1$, $j = 1,\ldots,i-1$, *have a zero at infinity.* □

Since sometimes overflows can occur in the intermediate stages of a Runge-Kutta method [14], Verwer [29] defined

DEFINITION 3 [29]. A Generalized Runge-Kutta method is said to be *Internally S-stable* if at each mth stage, $m = 1,\ldots,v$, the corresponding scheme of stage m is S-stable. If the above conditions only hold for $|\arg(-Z)| < \alpha$, then the method is said to be Internally $S(\alpha)$-stable. □

THEOREM 2 [29]. *A Generalized Runge-Kutta method is Internally S-stable if*
(a) $R^{(m)}(Z)$, $m = 1,\ldots,v$, *is Strongly A-acceptable, and*
(b) *The coefficient functions* A_{ij}, $i = 2,\ldots,v+1$, $j = 1,\ldots,i-1$, *have a zero at infinity.* □

Note that all the S-stable methods are strongly A-stable, but not vice versa.

3. Consistency. If we have $x_{n+1} = x_n + h_n$ and denote the solution $y(x_n)$ of Eq. (1.1) by y_n, and the kth derivative of $y(x)$ with respect to $x(d^k y/dx^k)$ by $y^{[k]}$, then the solution at x_{n+1} is given by the Taylor series

$$y_{n+1} = y_n + \sum_{j=1}^{\infty} 1/j!h_n^j y_n^{[j]}. \tag{3.1}$$

Runge-Kutta methods eliminate the need to calculate higher derivatives by using extra function evaluations. In order to derive Runge-Kutta processes, we start with the Taylor Series [21, p. 58]:

$$f(y+c) = f(y) + \sum_{j=1}^{\infty} 1/j(c\cdot\nabla)^j f(y), \tag{3.2}$$

where c is an increment not dependent on y and ∇ is the gradient operator.

LEMMA 1. *Equation* (3.2) *can be written in the equivalent form*

$$f(y+c) = \sum_{j=0}^{\infty} v^{(j)}, \tag{3.3}$$

where the terms $v^{(j)}$ *are given by the recurrence relation*

$$v^{(j)} = 1/j\partial/\partial y[v^{(j-1)}]c, \tag{3.4}$$

where $v^{(0)} = f(y)$.

Proof. The proof is by inspection, using Eqs. (3.2), (3.3), and (3.4). □

The Taylor series (3.3) thus has the form

$$\begin{aligned} f(y+c) = f(y) + Jc + 1/2!\partial/\partial y[Jc]c \\ + 1/3!\partial/\partial y(\partial/\partial y[Jc]c)c + \cdots, \end{aligned} \tag{3.5}$$

where $J \equiv J(y) \equiv \partial f(y)/\partial y =$ Jacobian matrix and the notation $\partial/\partial y[v(y)]$, for any vector v a function of y, is taken to represent a matrix whose ijth element is $\partial v_i/\partial y_j$. At this point we quote some results of Butcher [6], who obtained expressions for the derivatives $y^{[k]}$ of (3.1) in terms of "elementary differentials", which he defined. Butcher obtained general results. However, we will list the results for orders $p \leqslant 4$.

$$\begin{aligned} y_n^{[1]} = f,\quad y_n^{[2]} = \{f\},\quad y_n^{[3]} = \{{}_2f\}_2 + \{f^2\}, \\ y_n^{[4]} = \{{}_3f\}_3 + \{{}_2f^2\}_2 + 3\{\{f\}f\} + \{f^3\}. \end{aligned} \tag{3.6}$$

The elementary differentials on the right-hand side of Eqs. (3.6) are defined in [6]. Using this definition, and comparing the ith components of the vectors, we have the following identities:

$$
\begin{gathered}
f \equiv f_n, \quad \{f\} \equiv J_n f_n, \quad \{{}_2 f\}_2 \equiv J_n^2 f_n, \quad \{f^2\} \equiv \partial/\partial y[Jf_n]_n f_n, \\
\{{}_3 f\}_3 \equiv J_n^3 f_n, \quad \{{}_2 f^2\}_2 \equiv J_n \partial/\partial y[Jf_n]_n f_n, \\
\{\{f\}f\} \equiv \partial/\partial y[Jf_n]_n J_n f_n \equiv \partial/\partial y[JJ_n f_n]_n f_n, \\
\{f^3\} \equiv \partial/\partial y(\partial/\partial y[Jf_n] f_n)_n f_n.
\end{gathered} \tag{3.7}
$$

In the formulas (3.7), both J and f are functions of y, and the notation J_n and f_n are the values of J and f, respectively, evaluated at $y = y(x_n)$. The notation $\partial/\partial y[v]_n$, for any vector v a function of y, is the matrix $\partial v(y)/\partial y$ evaluated at $y = y(x_n)$. Hence, for example, we have

$$
\partial/\partial y[Jf_n]_n = \partial/\partial y[J(y(x))f(y(x_n))]_{x=x_n}.
$$

The vectors on the right-hand side of Eqs. (3.7) arise naturally when the Taylor series expansion of the form in Lemma 1 is used.

3.1. *Type* 1. From Lemma 1, any Taylor series expansion, which includes matrix functions of $\bar{J}_n$ (an approximate Jacobian), will give rise to vectors which include both J_n and $\bar{J}_n$ as arguments. Hence we will describe these elementary differentials as

$$
\begin{gathered}
\phi^{(2)} \equiv \bar{J}_n f_n, \quad \phi_1^{(3)} \equiv \bar{J}_n^2 f_n, \quad \phi_2^{(3)} \equiv J_n \bar{J}_n f_n, \quad \phi_3^{(3)} \equiv \bar{J}_n J_n f_n, \\
\phi_1^{(4)} \equiv \bar{J}_n^3 f_n, \quad \phi_2^{(4)} \equiv \bar{J}_n^2 J_n f_n, \quad \phi_3^{(4)} \equiv J_n \bar{J}_n^2 f_n, \\
\phi_4^{(4)} \equiv \bar{J}_n J_n \bar{J}_n f_n, \quad \phi_5^{(4)} \equiv \bar{J}_n J_n^2 f_n, \quad \phi_6^{(4)} \equiv J_n \bar{J}_n J_n f_n, \\
\phi_7^{(4)} \equiv J_n^2 \bar{J}_n f_n, \quad \phi_8^{(4)} \equiv \bar{J}_n \partial/\partial y[J_n f_n]_n f_n, \quad \phi_9^{(4)} \equiv \partial/\partial y[Jf_n]_n \bar{J}_n f_n.
\end{gathered} \tag{3.8}
$$

Thus, using Lemma 1 and Eqs. (3.6), (3.7), and (3.8), consistency conditions can be established for processes of order $p \leqslant 4$, although by using the results of Butcher [6] pertaining to higher orders, and extending Eqs. (3.7) and (3.8) to include higher order differentials, processes of higher order can be obtained.

We now need to know the minimum number of stages required to construct a Type 1 process of given order.

THEOREM 3. (a) *A Type* 1 *process of order p requires at least as many stages as an explicit method of the same order.*

(b) *Furthermore, the consistency equations of a v stage Type* 1 *process contain as a subset the consistency equations for a v stage explicit Runge-Kutta process of the same order.*

Proof. Since the process must remain consistent for all inaccurate Jacobians ($\bar{J}_n$), it must be consistent for the null matrix ($\bar{J}_n \equiv 0$). This gives an explicit method. Both results then follow. □

Steihaug and Wolfbrandt [27] were aware of the connection between the Type 1 methods and explicit methods, although they did not formally produce the lower bound on the attainable order of these processes for a given number of function evaluations, which follows from Theorem 3. Instead they obtained an upper bound, showing that if the denominator of the stability function (2.5) has order m, then the maximum attainable order is $m + 1$. For a v stage process, the methods examined by

Steihaug and Wolfbrandt [27] (which are a subset of the Generalized Runge-Kuttas, and are called Modified Rosenbrock methods) have maximum attainable order $(v+1)$. For the methods examined in this paper, however, the order of the polynomial in the denominator of (2.5) is usually greater than v, for a v stage processs, and thus the upper bound is rather conservative. It seems at this time that the lower bound given by Theorem 3 is likely to be more useful when designing Type 1 methods of given order.

3.2. *Type* 2. To obtain consistency for Type 2 processes, we expand the Jacobian $J_{n-k} \equiv \partial f(y(x))/\partial y\,|_{x=x_{n-k}}$ in the Taylor series

$$J_{n-k} = J_n + \sum_{j=1}^{\infty} 1/j!\,(-\mu_k h_n)^j J_n^{[j]}, \tag{3.9}$$

where $x_n = x_{n-k} + \mu_k h_n$, $x_{n+1} = x_n + h_n$, with

$$\mu_k = \sum_{j=1}^{k} h_{n-j}/h_n, \qquad k \geqslant 1, \mu_0 = 0, \tag{3.10}$$

and where $J_n^{[l]}$, $l = 1, 2, \ldots$, is a matrix in which the ijth element is

$$d^l/dx^l\big(\partial f_i(y(x))/\partial y_j\big)\,|_{x=x_n}.$$

That is, it is the lth derivative of the ijth element of J, evaluated at $y = y(x_n)$. From Lemma 1, it can be seen that a Type 2 process will contain terms such as $J_n^{[1]} f_n$, $J_n\, J_n^{[1]} f_n$, $J_n^{[1]} J_n f_n$, and $J_n^{[2]} f_n$. These can be related to the elementary differentials of Butcher in Eq. (3.6). By comparing the ith components of the vectors, we have the following identities

$$\begin{aligned} J_n^{[1]} f_n &\equiv \{f^2\}, \quad J_n J_n^{[1]} f_n \equiv \{{}_2 f^2\}_2, \\ J_n^{[1]} J_n f_n &\equiv \{\{f\}f\}, \quad J_n^{[2]} f_n \equiv \{\{f\}f\} + \{f^3\}, \end{aligned} \tag{3.11}$$

The presence of the parameter μ_k in (3.9) causes problems in obtaining consistency, since μ_k will increase from step to step (see (3.10)) until the Jacobian is reevaluated. From (3.11) it is apparent that the consistency equations will contain μ_k, although we will derive processes in which the *scalar parameters are not functions of* μ_k, in order to avoid the computation of these parameters at every step, and eliminate the possibility that these parameters become very small or very large, with consequent loss of significant figures of accuracy. In addition, we will obtain processes in which *the leading truncation error term is not a function of* μ_k, since this would mean the truncation error would increase rapidly from step to step with consequent frequent reevaluation of the Jacobian. This is because we are obliged to use the estimate of error to judge when a computation is becoming unstable, and hence reevaluate the Jacobian to stabilize the solution. Since the primary purpose of these processes is to avoid this chore as much as possible, we will accept possible reductions in attainable order so that this aim is accomplished. Note, however, that the second truncation error term will contain μ_k, and eventually this term will dominate the truncation error, so that it will not normally be possible to avoid reevaluating the Jacobian indefinitely.

THEOREM 4. *No second order one stage Type* 2 *process (in which the leading truncation error term is not a function of* μ_k*) exists.*

Proof. From Eq. (3.11) it is apparent that the Taylor series expansion for J_{n-k} gives rise to the elementary differential $\{f^2\}$. By inspection of (3.7) it is obvious that $\{f^2\}$ does not appear from any other source for a one stage process. It is thus apparent that the coefficient of $\{f^2\}$ for any second order one stage process gives $\mu_k = 1/3$, so that μ_k cannot be removed from the leading truncation error term. □

Thus, using Lemma 1, (3.6), (3.7) and (3.11), consistency conditions for Type 2 processes can be obtained.

4. Some Type 1 Processes. It is well known that, for an explicit method of order p ($p \leqslant 4$), we require p stages to obtain consistency (see, for example, [22, p. 120]). Hence we will construct a second order process with two stages and a third order process with three stages. The second order process has the general form

$$y_{n+1} = y_n + A_{31}k_1 + A_{32}k_2, \qquad k_1 = h_n f(y_n), \quad k_2 = h_n f(y_n + A_{21}k_1). \tag{4.1}$$

The characteristic matrix (2.4) has the form

$$\begin{bmatrix} 0 & 0 \\ \dfrac{a_{21}}{(1-bZ)} & 0 \\ \hline \dfrac{a_{31}(1+\alpha_{311}Z+\alpha_{312}Z^2)}{(1-bZ)^3} & \dfrac{a_{32}(1+\alpha_{321}Z)}{(1-bZ)^2} \end{bmatrix}. \tag{4.2}$$

Using the results of Section 3, we have, from the coefficients of the elementary differentials $f, \{f\}$, and $\phi^{(2)}$:

$$a_{31} + a_{32} = 1, \tag{4.3}$$

$$a_{32}a_{21} = 1/2, \tag{4.4}$$

$$a_{31}(3b + \alpha_{311}) + a_{32}(2b + \alpha_{321}) = 0. \tag{4.5}$$

Equations (4.3) and (4.4) are those of a second order, two stage explicit process, as Theorem 3 indicated. We select the solution

$$a_{31} = -1, \quad a_{32} = 2, \quad a_{21} = 1/4, \quad \alpha_{311} = -3b, \quad \alpha_{321} = -2b.$$

From Eq. (2.5) we have

$$R^{(1)}(Z) = [1 + (1/4 - b)Z]/(1 - bZ) \tag{4.6}$$

and

$$R^{(2)}(Z) = [1 + (1 - 3b)Z + (3b^2 - 3b + 1/2)Z^2]/(1 - bZ)^3 \tag{4.7}$$

if

$$a_{31}\alpha_{312} = b^3 - a_{32}\alpha_{321}(a_{21} - b), \tag{4.8}$$

i.e., $\alpha_{312} = -b^3 + 4b^2 - b$. For $b = 0.435\,866\,521\,508\,459$, which is a solution of the equation $b^3 - 3b^2 + 3b/2 - 1/6 = 0$, the stability functions $R^{(1)}(z)$ and $R^{(2)}(Z)$ are Strongly A-acceptable and L-acceptable, respectively, and thus, from Theorem 2, the process (4.2) is Internally S-stable. We choose this value of b, since it gives us a process which is second order in general, but third order when $\bar{J}_n = J_n =$ constant matrix [10].

The error estimator can be developed by deriving a first order formula of the same form as (4.2), so that we have the characteristic matrix

$$\left[\begin{array}{cc} 0 & 0 \\ \dfrac{a_{21}}{(1-bZ)} & 0 \\ \hline \dfrac{\bar{a}_{31}(1+\bar{\alpha}_{311}Z+\bar{\alpha}_{312}Z^2)}{(1-bZ)^3} & \dfrac{\bar{a}_{32}(1+\bar{\alpha}_{321}Z)}{(1-bZ)^2} \end{array}\right]. \tag{4.9}$$

The selection

$$\begin{gathered} \bar{a}_{31} = a_{31} + \delta = -1 + \delta, \quad \bar{a}_{32} = a_{32} - \delta = 2 - \delta, \\ \bar{a}_{31}\bar{\alpha}_{311} = a_{31}\alpha_{311} - b\delta = 3b - b\delta, \\ \bar{a}_{31}\bar{\alpha}_{312} = a_{31}\alpha_{312} = b^3 - 4b^2 + b, \\ \bar{a}_{32}\bar{\alpha}_{321} = a_{32}\alpha_{321} = -4b, \end{gathered} \tag{4.10}$$

gives a process which is first order with a truncation error

$$\varepsilon_{n+1} = \delta h_n^2 \phi^{(2)} + O(h_n^3), \tag{4.11}$$

where δ is an arbitrary constant. The stability function of the error estimator is

$$R^{(2)}(Z) = \big(1 + (1-3b)Z + (3b^2 - 3b + 1/2 + \delta)Z^2\big)/(1-bZ)^3.$$

Subtraction of the two solutions gives an estimate of the error of the lower order formula (4.11). Since a solution by the higher order formula is available with no extra computation, we accept this solution since, normally, it will be more accurate than the lower order formula. In particular, if the second order method has local truncation error

$$\varepsilon_{n+1}^{(2)} = \psi_1 h_n^3 + O(h_n^4),$$

then the first order method has error

$$\varepsilon_{n+1}^{(1)} = \delta h_n^2 \phi^{(2)} + h_n^3(\psi_1 + \delta\psi_2) + O(h_n^4),$$

so that our estimate of error of the second order method is

$$e_{\mathrm{EST}} = \delta h_n^2 \phi^{(2)} + \delta h_n^3 \psi_2 + O(h_n^4),$$

where ψ_1 and ψ_2 are functions of the elementary differentials. Thus we must select δ such that

$$e_{\mathrm{EST}} \simeq \varepsilon_{n+1}^{(2)} \quad \text{(i.e.)}\ \delta\big(h_n^2\phi^{(2)} + h_n^3\psi_2\big) \simeq h_n^3\psi_1.$$

If δ is too small, the error is underestimated, and if δ is too large, the error is overestimated, resulting in more computation than is needed to satisfy the error criterion. The problem of choosing a suitable δ is a difficult one, which probably can only be solved with extensive computational experience. In this paper we have chosen δ to be quite large, and hence conservative (see Section 7). If, however, we chose to accept the first order solution, then we would have had to select δ such that

$$\delta\big(h_n^2\phi^{(2)} + h_n^3\psi_2\big) \simeq \delta h_n^2\phi^{(2)} + h_n^3(\psi_1 + \delta\psi_2)$$

which, if the leading error terms dominate, is true for all δ. That is

$$\lim_{h\to 0} e_{\mathrm{EST}} = \varepsilon_{n+1}^{(1)}.$$

However, there are restrictions on the values of δ which can be used if the first order formula is accepted, since $R^{(2)}(Z)$ will not be L-acceptable if δ is too large. These restrictions do not apply if the second order solution is accepted, since δ does not affect the formula, only the estimate of error. In addition, as mentioned above, we prefer the higher order formula since it is normally more accurate than the lower order formula. The efficiency of the integration process can be improved by using the partial fraction expansions

$$\begin{aligned}(1 + \alpha_{311}Z + \alpha_{312}Z^2)/(1 - bZ)^3 \\ = (\alpha_{312}/b^2)/(1 - bZ) + (-\alpha_{311}/b - 2\alpha_{312}/b^2)/(1 - bZ)^2 \\ + (1 + \alpha_{311}/b + \alpha_{312}/b^2)/(1 - bZ)^3\end{aligned} \tag{4.12}$$

and

$$\begin{aligned}(1 + \alpha_{321}Z)/(1 - bZ)^2 \\ = (-\alpha_{321}/b)/(1 - bZ) + (1 + \alpha_{321}/b)/(1 - bZ)^2.\end{aligned} \tag{4.13}$$

Thus all the matrix * vector multiplications are eliminated. If we denote the matrix $B = [I - h_n b\bar{J}_n]$, then we can obtain the solution by

$$\begin{aligned}k_1 = h_n f(y_n), \quad k_2 = h_n f(y_n + 1/4B^{-1}k_1), \\ y_{n+1} = y_n + (\beta_1 B^{-1}k_1 + \beta_2 B^{-2}k_1 + \beta_3 B^{-3}k_1 + \beta_4 B^{-1}k_2 + \beta_5 B^{-2}k_2), \\ \varepsilon_{n+1} = \delta(\bar{\beta}_1 B^{-2}k_1 + \bar{\beta}_2 B^{-2}k_2),\end{aligned} \tag{4.14}$$

where

$$\begin{aligned}\beta_1 = b - 4 + 1/b, \quad \beta_2 = -3 - 2\beta_1, \quad \beta_3 = 2 + \beta_1, \\ \beta_4 = 4, \quad \beta_5 = -2, \quad \bar{\beta}_1 = 4, \quad \bar{\beta}_2 = -4,\end{aligned}$$

and $b = 0.435\,866\,521\,508\,459$. Thus the process is (2, 2, 5, 0).

In a similar manner, we can develop a third order process of the form

$$\begin{aligned}y_{n+1} = y_n + A_{41}k_1 + A_{42}k_2 + A_{43}k_3, \quad k_1 = h_n f(y_n), \\ k_2 = h_n f(y_n + A_{21}k_1), \quad k_3 = h_n f(y_n + A_{31}k_1 + A_{32}k_2).\end{aligned} \tag{4.15}$$

The characteristic matrix (2.4) has the form

$$\left[\begin{array}{ccc} 0 & 0 & 0 \\ \dfrac{a_{21}}{(1 - bZ)} & 0 & 0 \\ \dfrac{a_{31}(1 + \alpha_{311}Z + \alpha_{312}Z^2)}{(1 - bZ)^3} & \dfrac{a_{32}(1 + \alpha_{321}Z)}{(1 - bZ)^2} & 0 \\ \hline \dfrac{a_{41}(1 + \alpha_{411}Z + \alpha_{412}Z^2 + \alpha_{413}Z^3)}{(1 - bZ)^4} & \dfrac{a_{42}(1 + \alpha_{421}Z)}{(1 - bZ)^2} & \dfrac{a_{43}}{(1 - bZ)} \end{array}\right]. \tag{4.16}$$

The following solution satisfies the consistency equations:

$$\begin{aligned}a_{41} = 1/6, \quad a_{42} = 2/3, \quad a_{43} = 1/6, \quad a_{21} = a_{21} = 1/2, \quad a_{31} = -1, \quad a_{32} = 2, \\ \alpha_{411} = -3b, \quad \alpha_{412} = 9b^2, \quad \alpha_{421} = -5b/2, \quad \alpha_{311} = -b, \quad \alpha_{321} = -2b.\end{aligned}$$

The stability functions are

$$R^{(1)}(Z) = [1 + (1/2 - b)Z]/(1 - bZ),$$
$$R^{(2)}(Z) = [1 + (1 - 3b)Z + (3b^2 - 5b + 1)z^2]/(1 - bZ)^3,$$

if $\alpha_{321} = -b^3 + 4b^2 - 2b$, and

$$R^{(3)}(Z) = [1 + (1 - 4b)Z + (6b^2 - 4b + 1/2)Z^2 + (-4b^3 + 6b^2 - 2b + 1/6)Z^3]/(1 - bZ)^4, \tag{4.17}$$

if $b^4 + a_{41}\alpha_{413} - a_{42}\alpha_{421}b(1/2 - b) = 0$, i.e.,

$$\alpha_{413} = -6b^4 + 10b^3 - 5b^2. \tag{4.18}$$

For $b = 0.572\,816\,062\,5$, which is a solution of the equation

$$b^4 - 4b^3 + 3b^2 - 2b/3 + 1/24 = 0,$$

the stability function $R^{(1)}(Z)$ is strongly A-acceptable, $R^{(2)}(Z)$ is $L(\alpha)$-acceptable ($\alpha \in [0, 75°]$), and $R^{(3)}(Z)$ is L-acceptable, so that, from Theorems 1 and 2, the process (4.16) is S-stable and Internally $S(\alpha)$-stable for $\alpha \leqslant 75°$. We choose this value of b, since it gives us a process which is third order in general, but fourth order when $\bar{J}_n = J_n =$ constant matrix [10]. We construct an error estimator by deriving a second order formula of the same form as (4.16), and the error estimate is obtained in a similar manner to the second order process. Thus the process is given by

$$y_{n+1} = y_n + (\beta_6 B^{-1}k_1 + \beta_7 B^{-2}k_1 + \beta_8 B^{-3}k_1 + \beta_9 B^{-4}k_1 + \beta_{10}B^{-1}k_2 + \beta_{11}B^{-2}k_2 + \beta_{13}B^{-1}k_3), \tag{4.19}$$

where

$$B = [I - h_n b\bar{J}_n], \quad k_1 = h_n f(y_n), \quad k_2 = h_n f(y_n + 1/2B^{-1}k_1),$$
$$k_3 = h_n f(y_n + \beta_1 B^{-1}k_1 + \beta_2 B^{-2}k_1 + \beta_3 B^{-3}k_1 + \beta_4 B^{-1}k_2 + \beta_5 B^{-2}k_2),$$

and the error estimate is given by

$$\varepsilon_{n+1} = \delta(\bar{\beta}_1 B^{-1}k_1 + \bar{\beta}_2 B^{-2}k_1 + \bar{\beta}_3 B^{-3}k_1 + \bar{\beta}_4 B^{-4}k_1 + \bar{\beta}_5 B^{-1}k_2 + \bar{\beta}_6 B^{-1}k_3),$$

where

$$\begin{aligned} &\beta_1 = b - 4 + 2/b, \quad \beta_2 = -1 - 2\beta_1, \quad \beta_3 = \beta_1, \quad \beta_4 = 4, \quad \beta_5 = -2, \\ &\beta_6 = b - 5/3 + 5/(6b), \quad \beta_7 = 3/2 - 3\beta_6, \quad \beta_8 = -5/2 + 3\beta_6, \\ &\beta_9 = 7/6 - \beta_6, \quad \beta_{10} = 5/3, \quad \beta_{11} = -1, \quad \beta_{12} = 1/6, \quad \bar{\beta}_1 = 1/b - 2, \\ &\bar{\beta}_2 = -3 - 3\bar{\beta}_1, \quad \bar{\beta}_3 = -\bar{\beta}_2, \quad \bar{\beta}_4 = -1 - \bar{\beta}_1, \quad \bar{\beta}_5 = 2, \quad \bar{\beta}_6 = -1, \end{aligned} \tag{4.20}$$

and $b = 0.572\,816\,062\,5$. Thus the process is (3, 3, 7, 0).

It should be noted that the error estimator described in this section for second and third order processes is unlikely to be successful for higher order formulas. The consistency equations for an explicit method are a subset of the consistency equations for a Type 1 process, and so the problems of obtaining an error estimator are the same as for explicit Runge-Kuttas. Thus, for the higher order formulas, it might be more fruitful to generalize successful explicit methods such as the Fehlberg formulas [15], [16].

It should also be noted that in this section we have derived processes which are (2, 2, 5, 0) and (3, 3, 7, 0), respectively. It is possible to obtain processes which are (2, 2, 3, 0) and (3, 3, 6, 0), although in these cases the error estimator will have a stability function which is not asymptotically zero (i.e., $\lim_{Z\to-\infty}|R(Z)|=0$). We prefer the error estimator to have an asymptotically zero stability function, since it is easy to imagine situations where the stiff components of a system cause the error estimator to select a step size which is smaller than it has to be, from truncation error considerations. In addition, we prefer a method of order p to be order $(p+1)$ if $\bar{J}_n = J_n =$ a constant matrix, since the extra accuracy is obtained at only a modest increase in computation. This latter requirement gives rise to reference formulas which are asymptotically zero.

5. A Type 2 Process. As a consequence of Theorem 4, we will construct a third order, two stage process of the general form

$$y_{n+1} = y_n + A_{31}k_1 + A_{32}k_2, \quad k_1 = h_n f(y_n), \quad k_2 = h_n f(y_n + A_{21}k_1), \tag{5.1}$$

with characteristic matrix

$$\begin{bmatrix} 0 & 0 \\ \dfrac{a_{21}(1+\alpha_{211}Z)}{(1-bZ)^2} & 0 \\ \hline \dfrac{a_{31}(1+\alpha_{311}Z+\alpha_{312}Z^2+\alpha_{313}Z^3)}{(1-bZ)^4} & \dfrac{a_{32}(1+\alpha_{321}Z)}{(1-bZ)^2} \end{bmatrix}. \tag{5.2}$$

Using the results of Section 3, we have the consistency equations, which arise from the coefficients of f, $\{f\}$, $\{_2f\}_2$, $\{f^2\}$, $\{_3f\}_3$, $\{_2f^2\}_2$, $\{\{f\}f\}$, and $\{f^3\}$, respectively:

$$a_{31} + a_{32} = 1, \tag{5.3}$$

$$a_{31}(4b + \alpha_{311}) + a_{32}(2b + a_{21} + \alpha_{321}) = 1/2, \tag{5.4}$$

$$\begin{aligned} &a_{31}(10b^2 + 4b\alpha_{311} + \alpha_{312}) \\ &\quad + a_{32}[a_{21}(4b + \alpha_{211} + \alpha_{321}) + 3b^2 + 2b\alpha_{321}] = 1/6, \end{aligned} \tag{5.5}$$

$$1/2a_{32}a_{21}^2 - \mu_k[a_{31}(4b + \alpha_{311}) + a_{32}(2b + \alpha_{321})] = 1/6, \tag{5.6}$$

$$\begin{aligned} &a_{31}[20b^3 + 10b^2\alpha_{311} + 4b\alpha_{312} + \alpha_{313}] \\ &\quad + a_{32}[4b^3 + 3b^2\alpha_{321} + a_{21}(10b^2 + 4b\alpha_{211} + 4b\alpha_{321} + \alpha_{211}\alpha_{321})] = 1/24, \end{aligned} \tag{5.7}$$

$$\begin{aligned} &1/2a_{32}a_{21}^2(2b + \alpha_{321}) - \mu_k a_{31}(10b^2 + 4b\alpha_{311} + \alpha_{312}) \\ &\quad - \mu_k a_{32}[3b^2 + 2b\alpha_{321} + a_{21}(2b + \alpha_{211})] = 1/24, \end{aligned} \tag{5.8}$$

$$\begin{aligned} &a_{32}a_{21}^2(2b + \alpha_{211}) - \mu_k a_{31}(10b^2 + 4b\alpha_{311} + \alpha_{312}) \\ &\quad - \mu_k a_{32}[3b^2 + 2b\alpha_{321} + a_{21}(2b + \alpha_{321})] \\ &\quad + 1/2\mu_k^2[a_{31}(4b + \alpha_{311} + a_{32}(2b + \alpha_{321}))] = 1/8, \end{aligned} \tag{5.9}$$

$$1/6a_{32}a_{21}^3 + 1/2\mu_k^2[a_{31}(4b + \alpha_{311}) + a_{32}(2b + \alpha_{321})] = 1/24. \tag{5.10}$$

The solution

$$a_{31} = 1/4, \quad a_{32} = 3/4, \quad a_{21} = 2/3, \quad \alpha_{211} = 1/3 - 2b, \quad \alpha_{321} = 1/3 - 2b,$$
$$\alpha_{311} = -1 - 4b, \quad \alpha_{312} = 9b^2 + 2b - 2/3,$$
$$\alpha_{313} = -4b^4 + 6b^3 - 9b^2 + 8b/3 - 2/9,$$
$$b = 0.572\,816\,062\,5,$$

where b is a solution of the equation

$$b^4 - 4b^3 + 3b^2 - 2b/3 + 1/24 = 0,$$

gives a process which is third order in general, fourth order when $J_n = J_{n-k}$ = a constant matrix, and in which the $O(h_n^4)$ terms are devoid of μ_k. In fact the truncation error is given by

$$\varepsilon_{n+1} = h_n^4\left[1/24\{{}_2f^2\}_2 + 1/8\{\{f\}f\} + 1/24\{f^3\}\right] + O(h_n^5). \tag{5.11}$$

The stability functions are

$$R^{(1)}(Z) = \left[1 + (2/3 - 2b)Z + (b^2 - 4b/3 + 2/9)Z^2\right]/(1 - bZ)^2 \tag{5.12}$$

and

$$R^{(2)}(Z) = \left[1 + (1 - 4b)Z + (6b^2 - 4b + 1/2)Z^2 + (-4b^3 + 6b^2 - 2b + 1/6)Z^3\right]/(1 - bZ)^4, \tag{5.13}$$

which are Strongly A-acceptable and L-acceptable, respectively, giving a process which is Internally S-stable (for $J_{n-k} = J_n$).

In a similar manner to the previous section, we construct an error estimator by seeking a second order process of the form

$$\begin{bmatrix} 0 & 0 \\ \dfrac{a_{21}(1 + \alpha_{211}Z)}{(1 - bZ)^2} & 0 \\ \hline \dfrac{\bar{a}_{31}(1 + \bar{\alpha}_{311}Z + \bar{\alpha}_{312}Z^2 + \bar{\alpha}_{313}Z^3)}{(1 - bZ)^4} & \dfrac{\bar{a}_{32}(1 + \bar{\alpha}_{321}Z)}{(1 - bZ)^2} \end{bmatrix}. \tag{5.14}$$

We choose the solution

$$\bar{a}_{31} = a_{31} - \delta, \quad \bar{a}_{32} = a_{32} + \delta, \quad \bar{a}_{31}\bar{\alpha}_{311} = a_{31}\alpha_{311} + \delta(4b - 1),$$
$$\bar{a}_{31}\bar{\alpha}_{312} = a_{31}\alpha_{312} + \delta(10b/3 - 5b^2 - 2/9), \tag{5.15}$$
$$\bar{a}_{31}\bar{\alpha}_{313} = a_{31}\alpha_{313} + \delta(-2b^3 + 3b^2 - 8b/9 + 2/27),$$
$$\bar{a}_{32}\bar{\alpha}_{321} = a_{32}\alpha_{321} + \delta(1/3 - 2b),$$

with $b = 0.572\,816\,062\,5$, which gives a second order process with truncation error

$$\begin{aligned} \varepsilon_{n+1} &= \delta h_n^3\left[2/9\{{}_2f\}_2 + (2/9 + 2/3\mu_k)\{f^2\}\right] + O(h_n^4) \\ &= \delta h_n^3\left[2/9y_n^{[3]} + 2/3\mu_k\{f^2\}\right] + O(h_n^4), \end{aligned} \tag{5.16}$$

where δ is an arbitrary constant, and μ_k is given by (3.10). The stability function is

$$R^{(2)}(Z) = \left[1 + (1 - 4b)Z + (6b^2 - 4b + 1/2)Z^2 + (-4b^3 + 6b^2 - 2b + 1/6 + 2/9\delta)Z^3\right]/(1 - bZ)^4, \tag{5.17}$$

which is asymptotically zero. Using partial fraction expansions the solution is given by:

$$
\begin{aligned}
&B = [I - h_n b J_{n-k}], \quad k_1 = h_n f(y_n), \quad k_2 = h_n f\big(y_n + \beta_1 B^{-1} k_1 + \beta_2 B^{-2} k_1\big),\\
&y_{n+1} = y_n + \big(\beta_3 B^{-1} k_1 + \beta_4 B^{-2} k_1 + \beta_5 B^{-3} k_1\\
&\qquad + \beta_6 B^{-4} k_1 + \beta_7 B^{-1} k_2 + \beta_8 B^{-2} k_2\big), \qquad (5.18)\\
&\varepsilon_{n+1} = 2/9\delta h_n\big(\bar\beta_1 B^{-1} k_1 + \bar\beta_2 B^{-2} k_1 + \bar\beta_3 B^{-3} k_1\\
&\qquad + \bar\beta_4 B^{-4} k_1 + \bar\beta_5 B^{-1} k_2 + \bar\beta_6 B^{-2} k_2\big),
\end{aligned}
$$

where $b = 0.572\,816\,062\,5$, and

$$
\begin{gathered}
d_1 = 2/(9b) - 4/3, \quad d_2 = -b + 3/2 - 9/(4b) + 2/(3b^2) - 1/(18b^3),\\
d_3 = 9/4 + 1/(2b) - 1/(6b^2), \quad d_4 = -1 - 1/(4b), \quad d_5 = -3/2 + 1/(4b),\\
\beta_1 = -d_1, \quad \beta_2 = 2/3 + d_1, \quad \beta_3 = -d_2, \quad \beta_4 = d_3 + 3d_2,\\
\beta_5 = -d_4 - 2d_3 - 3d_2, \quad \beta_6 = 1/4 + d_2 + d_3 + d_4, \quad \beta_7 = -d_5,\\
\beta_8 = 3/4 + d_5, \quad d_6 = -2 + 3/b - 8/(9b^2) + 2/(27b^3),\\
d_7 = -5 + 10/(3b) - 2/(9b^2), \quad d_8 = 4 - 1/b, \quad d_9 = 1/(3b) - 2,\\
\bar\beta_1 = -d_6, \quad \bar\beta_2 = d_7 + 3d_6, \quad \bar\beta_3 = -d_8 - 2d_7 - 3d_6,\\
\bar\beta_4 = -1 + d_8 + d_7 + d_6, \quad \bar\beta_5 = -d_9, \quad \bar\beta_6 = 1 + d_9.
\end{gathered}
$$

Thus the process is (3, 2, 6, 0) and shows some improvement in the amount of work required per step for the (3, 3, 7, 0) Type 1 method. Note that from (5.16) and (5.18), the measured error is

$$
\varepsilon_{n+1} = 2/9\delta h_n^3\big[y_n^{[3]} + 3\mu_k\{f^2\}\big], \qquad (5.19)
$$

which is liable to increase from one step to the next, if the Jacobian is not reevaluated. This is a desirable property, since the truncation error of the third order method (or more correctly the second term in the truncation error) will also increase from step to step. If this is not accounted for in some way by the measured error, then the results may be inaccurate because the measured error is not representative of the true error.

6. Comparison With Other Methods. Eitelberg [12] has obtained a second order Type 1 method which requires three function evaluations and two matrix factorizations per step. The author was exploring the possibility of using a block diagonal matrix as an approximation to the Jacobian. All Type 1 methods permit an arbitrary approximation while maintaining consistency, and so this method is not competitive with the (2, 2, 5, 0) method described in this paper. Steihaug and Wolfbrandt [27] examined a class of generalized Runge-Kuttas, which they called "Modified Rosenbrock" methods, and produced a (2, 2, 4, 2) method which remains consistent with an approximate Jacobian. The method uses a third order reference formula to give an error estimator of the type due to England [13]. In this method, extra stages are required to obtain the higher order reference formula. The authors produced a second order, two stage process which is L-stable (and hence S-stable, though not Internally S-stable) when the Jacobian is accurately evaluated. In order to obtain the

reference formula of third order, they used an extra two function evaluations, which are saved if the step size remains unchanged. If the step size changes, the process becomes (2, 3, 6, 3), however the algorithms for Type 1 methods usually try to keep the step size constant for reasonable periods, so that the process is essentially (2, 2, 4, 2) for a good part of the integration. This (2, 2, 4, 2) method thus requires slightly more work than the (2, 2, 5, 0) method described in this paper and does not share the same level of stability. However, the error estimator is liable to be more efficient, since the reference formula is third order, rather than first order. The (3, 3, 7, 0) method described in this paper is more accurate, being of higher order, and consequently requires more work per step. It remains to be seen if England's method can be extended to higher orders for Generalized Runge-Kuttas, since the more stages used, the higher the order of the stability function, and the more difficult it is to obtain highly stable formulas. It seems likely that the Fehlberg imbedded formulas [15], [16] will be more suitable for generalization.

We are unaware of any work which has been done on Type 2 methods.

7. Numerical Examples. The described methods were programmed in FORTRAN [10] and run on the PDP10 computer at the Prentice Computer Centre at the University of Queensland. The computation was done in double precision (18 significant figures), and the algorithm was modelled on Watts' and Shampine's program, as described in [17, p. 134]. Matrix operations of factorization and back substitution were performed using subroutines DECOMP and SOLVE described in [17, p. 51]. A mixed absolute and relative error criterion was used to control the step size, such that the allowable error at x_{n+1} is

$$e_{n+1} = \varepsilon_{\mathrm{REL}}(|y_{n+1}| + |y_n|)/2 + \varepsilon_{\mathrm{ABS}}, \tag{7.1}$$

and all the elements of the computed error vector $|\varepsilon_{n+1}|$ (see (4.14)) must be less than or equal to the corresponding elements of e_{n+1}. For the second order method we compute

$$h_{n+1}/h_n = (\|e_{n+1}/\varepsilon_{n+1}\|_\infty)^{1/2} \tag{7.2}$$

and for the third order methods

$$h_{n+1}/h_n = (\|e_{n+1}/\varepsilon_{n+1}\|_\infty)^{1/3}. \tag{7.3}$$

In order to reduce Jacobian evaluations and matrix factorizations, we increase the step size only by a factor of 2, if (7.2) or (7.3) give $h_{n+1}/h_n \geqslant 2$. If $h_{n+1}/h_n < 1$, the error criterion has not been satisfied, and the current step must be recomputed. Normally we would halve the step size and recompute a step if $h_n/h_{n+1} > 1$. However, since the error constant (δ) is arbitrary, we allowed a certain amount of latitude in accepting steps which did not satisfy the error tolerance. In particular, if (7.2) gives $h_n/h_{n+1} > 2^{1/2}$, or (7.3) gives $h_n/h_{n+1} > 2^{1/3}$, the step size is reduced by a factor of 2, and the step recomputed. This means we allowed δ to vary to approximately half its preselected value. This procedure eliminates the need for recomputing a step size when the error tolerance is only slightly exceeded.

Although the Type 1 methods are designed to be consistent with any approximate Jacobian, for the purpose of comparison with the Type 2 methods, we evaluate, for both classes, an accurate Jacobian which is reevaluated every time the step size is

changed, or after a specified number of steps. It follows that every time the Jacobian is evaluated, the matrix must be refactorized. The reason for reevaluating when there is a step change is that there is obviously a substantial change in the principal error function (if $h_{n+1}/h_n = 2$ or $= 1/2$) and hence possibly a change in the Jacobian, and caution is justified. The reason for reevaluating after a specified number of steps is that the truncation error is a function of the Jacobian ($\bar{J}_n$ or J_{n-k}) which if not reevaluated, tends to make the principal error function fairly constant in some cases, resulting in long periods of computations at small step sizes. It is desirable to purge the truncation error of the $\bar{J}_n$ or J_{n-k} terms at regular intervals. Tests were done with the number of steps per Jacobian evaluation being 1, 5, and unbounded, in order to illustrate this.

The initial step sizes are computed from, for the second order method:

$$h_0 = (\varepsilon_{\text{ABS}}/\| J_0 f_0 \|_\infty)^{1/2} \tag{7.4}$$

and for the third order method:

$$h_0 = (\varepsilon_{\text{ABS}}/\| J_0^2 f_0 \|_\infty)^{1/3} \tag{7.5}$$

with the maximum allowable value of $h_0 = 10^{-3}$ to avoid the situation where the principal error function might be very small at the $x = x_0$ (giving $h_0 \gg 0$), but increases very rapidly for $x > x_0$. Thus, chatter in the initial step size selection is avoided. The error constant (δ) was taken to be $\delta = 1$ for the Type 1 (second order) method, and $\delta = 1/2$ for the Type 1 (third order) and Type 2 (third order) methods. These values are most likely very conservative.

The algorithms were tested on two stiff systems, using error tolerances of $\varepsilon_{\text{REL}} = 10^{-4}$ and $\varepsilon_{\text{ABS}} = 10^{-8}$. The systems are taken from [28] and are

$$\begin{aligned} \text{I}\quad & y_1' = 0.01 - [1 + (y_1 + 1000)(y_1 + 1)][0.01 + y_1 + y_2], \\ & y_2' = 0.01 - [1 + y_2^2][0.01 + y_1 + y_2], \\ & y_1(0) = y_2(0) = 0. \end{aligned}$$

The reference solution is [28]

$$y_1(100) = -0.99164207, \quad y_2(100) = 0.9833636.$$

$$\begin{aligned} \text{II}\quad & y_1' = 0.04 - 0.04(y_1 + y_2) - 10^4 y_1 y_2 - 3 * 10^7 y_1^2, \\ & y_2' = 3 * 10^7 y_1^2, \\ & y_1(0) = y_2(0) = 0. \end{aligned}$$

The reference solution is [28]

$$y_1(10) = 0.1623391063 * 10^{-4}, \quad y_2(10) = 0.1586138424.$$

The first system is moderately stiff, and the solution over much of $x \in [0, 100]$ is almost a straight line, permitting an accurate solution by low order methods with large step sizes, providing the method is highly stable. The second system is very stiff and has the reputation [28] of being a severe test of an integration method. The results of the computation are given in Table 1. The parameter sd_j [28] is the number of significant digits which have been accurately computed at the reference point

$$\text{sd}_j = -\log | 1 - y_j/y_{j\text{REF}} | . \tag{7.6}$$

TABLE 1

Type/ Order	System	Steps per Jacob. Eval.	Steps	sd_1	sd_2	Funct. Eval.	Jacob. Eval.	Matrix Fact.	Back Subs.
1/2	I	1	472	4.2	4.4	949	477	477	2862
		5	469	3.7	3.7	943	111	111	2481
		∞	468	3.3	3.2	941	29	29	2394
	II	1	483	5.8	5.5	968	485	485	2910
		5	483	5.3	5.6	968	105	105	2530
		∞	930	2.9	3.5	1862	15	15	4675
1/3	I	1	99	4.6	4.9	307	104	104	832
		5	101	4.1	4.6	319	41	41	804
		∞	102	4.1	4.6	322	31	31	801
	II	1	93	5.0	6.4	281	94	94	752
		5	107	3.9	7.3	367	63	63	973
		∞	113	3.9	7.2	385	59	59	1011
2/3	I	1	69	3.7	3.8	143	74	74	518
		5	75	3.5	3.5	157	35	35	527
		∞	78	3.5	3.5	163	28	28	538
	II	1	69	4.6	5.9	140	71	71	497
		5	79	3.8	4.8	169	40	40	580
		∞	94	4.4	4.6	201	39	39	681

The effect of having the number of steps per Jacobian evaluated unbounded is apparent from the Type 1/order 2, System II test, and the Type 2/order 3, System II test.

8. Conclusions. The Type 1 processes are primarily useful when no accurate evaluation of a Jacobian is available. However, if the system is large, then the evaluation of a Jacobian might also be expensive, in comparison with function evaluations, and these methods might be more efficient than the usual Generalized Runge-Kuttas [29] which have fewer stages (one stage for a second order method, and two stages for a fourth order method [10]) but require a Jacobian evaluation at every step. This is especially true if the matrix has no special structure and must then require $n^3/3$ operations for factorization. The extra function evaluations required to obtain a Type 1 method is then not significant. The Type 2 methods require an accurate evaluation of a Jacobian and appear suitable for large systems where both Jacobian and function evaluations are expensive. It seems likely that they will require fewer function evaluations than a Type 1 method of the same order, although probably they will require more stages than the usual generalized Runge-Kuttas, because of the need to eliminate the parameter μ (see (3.10)) from the leading error term. However, the Type 2 methods seem superior to the Type 1 methods in terms of efficiency (see Table 1) for the particular algorithms used. However, another option not explored in this paper is to approximate the Jacobian

by some matrix having special structure (e.g., block diagonal, tridiagonal, etc.) which permits a cheap factorization. This approach was taken by Eitelberg [12]. Type 1 methods permit this approach, and Type 2 methods do not. Further development of both methods and mathematical software, with more extensive testing, is required. Furthermore, some results on the stability of Type 1 and Type 2 methods using approximate Jacobians (or Jacobians evaluated at a previous step) would be welcome.

9. Acknowledgement. We would like to thank an anonymous reviewer whose comments improved the presentation of this paper.

Department of Mechanical Engineering
University of Queensland
St. Lucia 4067, Queensland, Australia

1. R. H. ALLEN & C. POTTLE, "Stable integration methods for electronic circuit analysis with widely separated time constant," *Proc. Sixth Allerton Conf. on Circuit and System Theory*, IEEE, New York, 1968, pp. 311–320.

2. S. S. ARTEM'EV, "The construction of semi-implicit Runge Kutta methods," *Soviet Math. Dokl.*, v. 17, 1976, pp. 802–805.

3. S. S. ARTEM'EV & G. V. DEMIDOV, "A stable method for the solution of the Cauchy problem for stiff systems of ordinary differential equations," *Proc. 6th IFIP Conf. on Optimization Techniques*, Springer-Verlag, New York, 1975, pp. 270–274.

4. T. D. BUI, "On an *L*-stable method for stiff differential equations," *Inform. Process. Lett.*, v. 6, 1977, pp. 158–161.

5. T. D. BUI & S. S. GHADERPAVAH, "Modified Richardson Extrapolation scheme for error estimate in implicit Runge Kutta procedures for stiff systems of ordinary differential equations," *Proc. Seventh Manitoba Conf. on Numer. Math. and Computing*, Utilitas Mathematica Publishing, Inc., Winnipeg, 1977, pp. 251–268.

6. J. C. BUTCHER, "Coefficients for the study of Runge Kutta processes," *J. Austral. Math. Soc.*, v. 3, 1963, pp. 185–201.

7. J. C. BUTCHER, "Implicit Runge-Kutta processes," *Math. Comp.*, v. 18, 1964, pp. 50–64.

8. D. A. CALAHAN, "A stable, accurate method of numerical integration for nonlinear systems," *Proc. IEEE*, v. 56, 1968, p. 744.

9. J. R. CASH, "Semi-implicit Runge Kutta procedures with error estimates for the numerical integration of stiff systems of ordinary differential equations," *J. Assoc. Comput. Mach.*, v. 23, 1976, pp. 455–460.

10. J. D. DAY, *On Generalized Runge Kutta Methods*, Ph. D. Thesis, Dept. of Mechanical Engineering, Univ. of Queensland, 1980.

11. B. L. EHLE & J. D. LAWSON, "Generalized Runge Kutta processes for stiff initial value problems," *J. Inst. Math. Appl.*, v. 16, 1975, pp. 11–21.

12. E. EITELBERG, "Numerical simulation of stiff systems with a diagonal splitting method," *Math. Comput. Simulation*, v. 21, 1979, pp. 109–15.

13. R. ENGLAND, "Error estimates for Runge Kutta type solutions to systems of ordinary differential equations," *Comput. J.*, v. 12, 1969, pp. 166–170.

14. W. H. ENRIGHT, T. E. HULL & B. LINDBERG, "Comparing numerical methods for stiff systems of ODE's," *BIT*, v. 15, 1975, pp. 10–48.

15. E. FEHLBERG, *Classical Fifth-, Sixth-, Seventh- and Eighth-Order Runge Kutta Formulas With Step Size Control*, NASA Tech. Report R-287, 1968.

16. E. FEHLBERG, *Low Order Classical Runge Kutta Formulas With Step Size Control and Their Application to Some Heat Transfer Problems*, NASA Tech. Report R-315, 1969.

17. G. E. FORSYTHE, M. A. MALCOLM & C. B. MOLER, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, N. J., 1977.

18. C. W. GEAR, "Algorithm 407, DIFSUB for solution of ordinary differential equations," *Comm. ACM*, v. 14, 1971, pp. 185–190.

19. C. F. HAINES, "Implicit integration processes with error estimate for the numerical solution of differential equations," *Comput. J.*, v. 12, 1969, pp. 183–187.

20. P. J. VAN DER HOUWEN, *Explicit and Semi-Implicit Runge Kutta Formulas for the Integration of Stiff Equations*, Report TW132, Mathematisch Centrum, Amsterdam, 1972.

21. P. J. VAN DER HOUWEN, *Construction of Integration Formulas for Initial Value Problems*, North-Holland, Amsterdam, 1977.

22. J. D. LAMBERT, *Computational Methods in Ordinary Differential Equations*, Wiley, New York, 1973.

23. J. D. LAWSON, "Generalized Runge Kutta processes for stable systems with large Lipschitz constants," *SIAM J. Numer. Anal.*, v. 4, 1967, pp. 372–380.

24. S. P. NORSETT & A. WOLFBRANDT, "Order conditions for Rosenbrock type methods," *Numer. Math.*, v. 32, 1979, pp. 1–15.

25. A. PROTHERO & A. ROBINSON, "On the stability and accuracy of one step methods for solving stiff systems of ordinary differential equations," *Math. Comp.* v. 28, 1974, p. 145–162.

26. H. H. ROSENBROCK, "Some general implicit processes for the numerical solution of differential equations," *Comput. J.*, v. 5, 1963, pp. 329–330.

27. T. STEIHAUG & A. WOLFBRANDT, "An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations," *Math. Comp.*, v. 33, 1979, pp. 521–534.

28. J. G. VERWER, *Generalized Linear Multistep methods* II: *Numerical applications*, Report NW 12/75, Mathematisch Centrum, Amsterdam, 1975.

29. J. G. VERWER, "*S*-stability properties for generalized Runge Kutta methods," *Numer. Math.*, v. 27, 1977, pp, 359–370.