

ON THE IMPLEMENTATION OF SINGLY IMPLICIT RUNGE-KUTTA METHODS

G. J. COOPER

ABSTRACT. A modified Newton method is often used to solve the algebraic equations that arise in the application of implicit Runge-Kutta methods. When the Runge-Kutta method has a coefficient matrix A with a single point spectrum (with eigenvalue λ), the efficiency of the modified Newton method is much improved by using a similarity transformation of A . Each iteration involves vector transformations. In this article an alternative iteration scheme is obtained which does not require vector transformations and which is simpler in other respects also. Both schemes converge in a finite number of iterations when applied to linear systems of differential equations, but the new scheme uses the nilpotency of $A - \lambda I$ to achieve this. Numerical results confirm the predicted convergence for nonlinear systems and indicate that the scheme may be a useful alternative to the modified Newton method for low-dimensional systems. The scheme seems to become less effective as the dimension increases. However, it has clear advantages for parallel computation, making it competitive for high-dimensional systems.

1. INTRODUCTION

Two general approaches to the implementation of implicit Runge-Kutta methods have been proposed. In one approach, described by Chipman [8], a modified Newton method is used and schemes are developed to solve the resulting linear equations efficiently. Bickart [1] and Collings and Tee [9] describe schemes of this type. In another scheme of this general type, due to Butcher [5], a similarity transformation of the coefficient matrix of the method is used. This is particularly effective when the matrix has a single point spectrum, even though each step of the modified Newton iteration requires up to three vector transformations. Suppose that an s -stage method is applied to a system of n differential equations. Then each vector transformation requires $O(s^2n)$ operations. When the matrix of coefficients does not have a single point spectrum, Enright [12] suggested the use of an additional similarity transformation to transform the Jacobian of the differential system to Hessenberg form. This scheme is comparatively inefficient because each of the corresponding vector transformations requires $O(s^2n + sn^2)$ operations. In addition, problems arise when the coefficient matrix has a complex spectrum, although Varah [17] has suggested the

Received May 23, 1988; revised July 9, 1990.

1980 *Mathematics Subject Classification* (1985 Revision). Primary 65L05.

©1991 American Mathematical Society
0025-5718/91 \$1.00 + \$.25 per page

use of complex arithmetic. In a final proposal of this first general type, Cash [7] restricts attention to Runge-Kutta methods of a special form which makes the system of equations effectively of lower dimension. As in the approaches of Bickart and of Collings and Tee, the necessity to compute polynomials in the Jacobian matrix may aggravate ill-conditioning. For these reasons the properties of singly implicit methods have received much attention.

The other general approach is to use schemes based directly on iterative procedures. Frank and Ueberhuber [13] describe the use of iterated defect corrections. A variety of other iteration schemes have been discussed by Butcher [6]. Cooper and Butcher [10] considered a generalization of one of these schemes in which the Jacobian is used explicitly. They examined the convergence rate for a linear test problem and pointed out that superlinear convergence can be achieved only for singly implicit methods. A simple iteration scheme of this type is described in this article and the convergence rate for nonlinear systems is analyzed. Because the scheme does not require vector transformations in each step of the iteration, and because of the low computational overheads, the scheme may compete with the modified Newton method when n is small. Some numerical results are given to illustrate this and to confirm the convergence rate. The new scheme seems to be more efficient for parallel computation, since each step of the scheme consists of a number of independent stages. This advantage becomes more marked as n increases.

2. BACKGROUND

Consider the initial value problem for a system of n differential equations

$$x' = f(x), \quad x(t_0) = x_0,$$

and let x_r be an approximation to $x(t_r)$, $t_r = t_0 + rh$, $r = 1, 2, 3, \dots$. An s -stage Runge-Kutta method gives

$$x_{r+1} = x_r + h \sum_{i=1}^s b_i f(y_i),$$

where y_1, y_2, \dots, y_s satisfy the sn equations

$$(2.1) \quad x_r - y_i + h \sum_{j=1}^s a_{ij} f(y_j) = 0, \quad i = 1, 2, \dots, s.$$

This article deals with singly implicit methods suitable for stiff systems, so that, although the matrix of coefficients $A = [a_{ij}]$ is not strictly lower triangular, there is a real nonsingular matrix S such that

$$(2.2) \quad S^{-1}AS = \lambda I + T,$$

where T is strictly lower triangular and I is the $s \times s$ identity matrix.

Let $Y = y_1 \oplus y_2 \oplus \dots \oplus y_s$ be an sn -element column vector, and let $X_r = x_r \oplus x_r \oplus \dots \oplus x_r$ and $F(Y) = f(y_1) \oplus f(y_2) \oplus \dots \oplus f(y_s)$. With this notation,

equations (2.1) may be represented as $D(Y) = 0$ with

$$(2.3) \quad D(Y) = X_r - Y + h(A \otimes I)F(Y),$$

where $A \otimes I$ is the direct product of A with the $n \times n$ identity matrix and, in general,

$$A \otimes R = \begin{bmatrix} a_{11}R & a_{12}R & \cdots & a_{1s}R \\ a_{21}R & a_{22}R & \cdots & a_{2s}R \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1}R & a_{s2}R & \cdots & a_{ss}R \end{bmatrix}.$$

The equation $D(Y) = 0$ may be solved by Newton iteration, noting that the derivative of D is given by $D'(Y) = h(A \otimes I)F'(Y) - I$ and that the derivative of F at Y is the block diagonal matrix $F'(Y) = f'(y_1) \oplus f'(y_2) \oplus \cdots \oplus f'(y_s)$, where each block is the derivative of f , the Jacobian evaluated at one of the points y_1, y_2, \dots, y_s . The Newton iteration is expensive because, in each step, the Jacobian is evaluated s times and a set of sn linear equations has to be solved.

In the modified Newton iteration, the Jacobian is computed comparatively infrequently. Let J be the Jacobian evaluated at some recent point x_p . The modified iteration scheme evaluates $\Delta^1, \Delta^2, \Delta^3, \dots$, and hence Y^1, Y^2, Y^3, \dots , to satisfy the equations

$$(2.4) \quad \begin{aligned} [I - hA \otimes J]\Delta^m &= D(Y^{m-1}), \\ Y^m &= Y^{m-1} + \Delta^m, \end{aligned} \quad m = 1, 2, 3, \dots$$

In each step of this iteration a set of sn linear equations has to be solved, but this work may be reduced by applying the transformation (2.2). The iteration scheme (2.4) becomes

$$(2.5) \quad \begin{aligned} [I - h(\lambda I + T) \otimes J]E^m &= (S^{-1} \otimes I)D(Y^{m-1}), \\ Y^m &= Y^{m-1} + (S \otimes I)E^m, \end{aligned} \quad m = 1, 2, 3, \dots,$$

and, since T is strictly lower triangular, each step requires the solution of s sets of n linear equations. The transformations $D(Y^{m-1}) \rightarrow (S^{-1} \otimes I)D(Y^{m-1})$ and $E^m \rightarrow (S \otimes I)E^m$ each require $O(s^2n)$ operations. The transformation $E^m \rightarrow (T \otimes J)E^m$ requires $O(s^2n + sn^2)$ operations but this work may be eliminated by expressing the iteration scheme (2.5) in the form

$$(2.6) \quad \begin{aligned} [I - h\lambda I \otimes J]E^m &= (CS^{-1} \otimes I)D(Y^{m-1}) - (L \otimes I)E^m, \\ Y^m &= Y^{m-1} + (S \otimes I)E^m, \end{aligned} \quad m = 1, 2, 3, \dots,$$

where $C = \lambda(\lambda I + T)^{-1}$ and $L = C - I$ is strictly lower triangular. Noting that $L = \lambda S^{-1}A^{-1}S - I$, we choose S so that $\lambda S^{-1}A^{-1}S$ is the Jordan canonical form of λA^{-1} . Then all elements of L are zero except for units on the

subdiagonal. In general, this subdiagonal is full, so that each step of the scheme (2.6) consists of s stages which have to be performed in sequence.

3. THE ITERATION SCHEME

Consider an iteration scheme of the form

$$(3.1) \quad \begin{aligned} [I - h\lambda I \otimes J]E^m &= (B \otimes I)D(Y^{m-1}), \\ Y^m &= Y^{m-1} + E^m, \end{aligned} \quad m = 1, 2, 3, \dots,$$

where B is a real nonsingular $s \times s$ matrix which is to be chosen so that the calculation of $(B \otimes I)D(W)$ is no more expensive than the calculation of $D(W)$. The aim is to obtain a scheme with rapid convergence which, because of the reduced computational cost of each step, can compete with the modified Newton iteration. Note that each step of the proposed scheme (3.1) consists of s stages which can be performed in parallel.

In the following it is assumed that the sequence $\{Y^m\}$ has limit Y , with the continuity of f ensuring that $D(Y) = 0$. Stronger conditions on f are needed to discuss the rate of convergence. Let f be continuously differentiable with derivative $f': \mathbb{R}^n \rightarrow \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$, where $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$ is the space of continuous linear maps of \mathbb{R}^n into \mathbb{R}^n with norm induced by a given norm on \mathbb{R}^n . Let f' satisfy the Lipschitz condition

$$\|f'(w) - f'(z)\| \leq L\|w - z\| \quad \forall w, z \in \mathbb{R}^n.$$

Now let \mathbb{R}^N , where $N = ns$, be normed by

$$\|V\| = \max_{1 \leq i \leq s} \|v_i\|, \quad V = v_1 \oplus v_2 \oplus \dots \oplus v_s.$$

This norm induces a norm on $\mathcal{L}(\mathbb{R}^N, \mathbb{R}^N)$, and F' satisfies a Lipschitz condition on \mathbb{R}^N with the same Lipschitz constant L . For elements in $\mathcal{L}(\mathbb{R}^N, \mathbb{R}^N)$ of the special form $B \otimes R$, where $B = [b_{ij}]$ is any $s \times s$ matrix and R any $n \times n$ matrix, it can be shown that $\|B \otimes R\| \leq \|B\| \|R\|$ with

$$\|B\| = \max_{1 \leq i \leq s} \sum_{j=1}^s |b_{ij}|.$$

To examine the rate of convergence of the iteration schemes, let $G(W) = F(Y) - F(W) - (I \otimes J)(Y - W)$ and note that, since $D(Y) = 0$,

$$(3.2) \quad D(W) = [I - hA \otimes J](Y - W) - h(A \otimes I)G(W).$$

Since $I \otimes J = F'(X_p)$, the identity

$$G(W) = \int_0^1 \{F'(W + t[Y - W]) - F'(X_p)\} dt[Y - W]$$

and the Lipschitz condition on F' give the inequality

$$(3.3) \quad \|G(W)\| \leq \frac{L}{2} \{\|Y - X_p\| + \|X_p - W\|\} \|Y - W\|.$$

For linear problems, $L = 0$. In an analysis of the Newton iteration, $I \otimes J$ would be replaced by $F'(W)$, giving $\|G(W)\| \leq \frac{L}{2}\|Y - W\|^2$. Although this second-order bound does not hold for the iteration scheme (3.1) or for the modified Newton iteration, $I \otimes J$ is often an appreciably better approximation to $F'(W)$ than is indicated by (3.3). On the other hand, for nonlinear systems, the bound (3.3) is replaced by a second-order bound only when the Jacobian is updated after each iteration and, in practice, only when the starting value is estimated sufficiently accurately.

Now consider the iteration scheme (3.1), and define $V^m = Y - Y^m$ for $m = 0, 1, 2, \dots$. Let $\bar{h} = \lambda h$ and $A = \lambda \bar{A}$, and apply (3.2) to give

$$(3.4) \quad V^m = [(I - B) \otimes I + \bar{h}(B\bar{A} - I) \otimes J][I \otimes (I - \bar{h}J)^{-1}]V^{m-1} + G^{m-1},$$

where $G^{m-1} = \bar{h}[(B\bar{A}) \otimes (I - \bar{h}J)^{-1}]G(Y^{m-1})$ for $m = 1, 2, 3, \dots$. To obtain rapid convergence, B should be chosen so that the spectral radius of

$$[(I - B) \otimes I + \bar{h}(B\bar{A} - I) \otimes J][I \otimes (I - \bar{h}J)^{-1}]$$

is zero for arbitrary J . One possibility is to choose $B = I$, but it is possible instead to choose $B = B(\alpha)$ so that $B\bar{A} - I = \alpha(I - B)$, giving $B = (1 + \alpha)(\bar{A} + \alpha I)^{-1}$ for $\alpha \neq -1$. With this choice,

$$(3.5) \quad [(I - B) \otimes I + \bar{h}(B\bar{A} - I) \otimes J][I \otimes (I - \bar{h}J)^{-1}] = M \otimes K,$$

where $M = I - B$ and K are given by

$$(3.6) \quad M = (\bar{A} + \alpha I)^{-1}(\bar{A} - I),$$

$$(3.7) \quad K = (I + \alpha \bar{h}J)(I - \bar{h}J)^{-1}.$$

This transformation (3.7) maps eigenvalues of $\bar{h}J$ into eigenvalues of K , and in the degenerate case, $\alpha = 1$, eigenvalues in the left half plane are mapped into eigenvalues of K of modulus at most one. This value of α seems to be particularly suitable for a general purpose iteration scheme.

Since the spectral radius of $\bar{A} - I$ is zero, (3.6) shows that the spectral radius of M , and hence of $M \otimes K$, is zero also. Note that M is an $s \times s$ matrix, so that $M^s = 0$. It now follows from (3.4) and (3.5) that

$$(3.8) \quad V^{m+s} = (M^{s-1} \otimes K^{s-1})G^m + \dots + (M \otimes K)G^{m+s-2} + G^{m+s-1},$$

$m = 0, 1, 2, \dots,$

so that, for linear systems, the iteration scheme terminates after s iterations. For nonlinear systems the performance of the scheme depends, in part, on the quality of $I \otimes J$ as an approximation to the derivative of F .

4. A COMPARISON WITH THE MODIFIED NEWTON ITERATION

In this section the iteration scheme (3.1) is compared with the modified Newton iteration. Linear algebra costs are compared in terms of operational counts for each iteration and in terms of the relative number of iterations needed by the two schemes.

Consider a typical step of the iteration scheme (3.1) and suppose that, at the beginning of the iteration, vectors $X_r - W$ and $\bar{h}F(W)$ have been computed. To transform $D(W) = X_r - W + h(A \otimes I)F(W)$ to $(B \otimes I)D(W)$, note that $B\bar{A} = \alpha B + (I + \alpha)I$, so that

$$(4.1) \quad (B \otimes I)D(W) = (B \otimes I)[X_r - W + \alpha\bar{h}F(W)] + (1 + \alpha)\bar{h}F(W).$$

When $\alpha = 1$, in particular, the scalar multiplications may be neglected and the transformation (4.1) requires s^2n multiplications, the same number needed to form $D(W)$. In the remainder of the step, s sets of n linear equations have to be solved. Suppose that the matrix $I - \bar{h}J$ has been factored as a product of a lower and upper triangular matrix. Then the solution of these s sets of equations requires sn^2 multiplications, giving a total of $s^2n + sn^2$ for one iteration.

Now consider the modified Newton iteration, implemented according to (2.6). Suppose that, at the beginning of a typical step, vectors $X_r - W$ and $hF(W)$ have been computed. Since there is no simplification of $(CS^{-1} \otimes I)D(W)$ corresponding to (4.1), the computation of this vector involves $2s^2n$ multiplications. Taking into account the solution of the s sets of linear equations and the final transformation, a total of $3s^2n + sn^2$ multiplications is needed for each iteration.

In comparing the two schemes it is necessary also to consider the number of iterations required. For a given problem, let ν be the number of iterations taken by the modified Newton iteration to achieve a specified accuracy, and let σ be the corresponding number of iterations taken by the scheme (3.1). On the basis of linear algebra costs only, the scheme (3.1) is competitive if

$$(4.2) \quad \rho \lesssim \frac{2s}{s+n}, \quad \text{where } \rho = \frac{\sigma - \nu}{\nu},$$

so that ρ , the relative increase in the number of iterations taken, needs to be small. One way to estimate ρ is to compare (3.8) with the corresponding result

$$(4.3) \quad V^{m-1} = h(A \otimes I)((I - hA \otimes J)^{-1}G(Y^m)), \quad m = 0, 1, 2, \dots,$$

for the modified Newton iteration (2.4). For linear problems, $\rho = (s-1)/2$ because each scheme requires one extra iteration to terminate the scheme. Although a comparison of (3.8) and (4.3) suggests that this is a good estimate of ρ for nonlinear problems also, this is not the case. Numerical evidence given in the next section shows that, while ρ increases with s , typically $\rho < 1$, and for some methods, ρ is substantially less than one. For such methods (4.2) indicates that the scheme (3.1) competes with the modified Newton iteration for low-dimensional problems. Note that an evaluation of F is required in each iteration, so there is an additional cost associated with (3.1). In part this can be offset against the higher overheads associated with the implementation of (2.6).

This comparison assumes sequential computation. For parallel computation there is a clear advantage in using the scheme (3.1), irrespective of the dimension

of the problem. In this scheme the s sets of linear equations can be solved in parallel. The transformation (4.1) can also be carried out using n sets of s^2 parallel multiplications. Suppose, for simplicity, that no other use is made of parallel computation. Then one iteration requires $n + n^2$ sets of multiplications performed in sequence.

In the modified Newton iteration the transformations also may be carried out using parallel multiplications, but the s sets of linear equations have to be solved in sequence. Thus, $3n + sn^2$ sets of multiplications are performed in sequence. Since $\rho \ll s$, there is a distinct advantage in using the scheme (3.1), an advantage which becomes more marked as n increases.

5. NUMERICAL RESULTS

In order to evaluate the efficiency of the iteration scheme suggested here, a number of numerical experiments were carried out and some results for three initial value problems are reported here. Although the Jacobians have very different eigensystems, the performance observed is similar and typical of tests with other problems not specifically reported. The results reported deal only with the case $\alpha = 1$ for the iteration scheme (3.1). A wide range of tests have shown that this choice is to be preferred.

By Problem 1 is meant the system

$$\begin{aligned}x_1' &= x_2, & x_1(0) &= 2, \\x_2' &= 5(1 - x_1^2)x_2 - x_1, & x_2(0) &= 0,\end{aligned}$$

obtained from the Van der Pol equation given by Davis [11]. The eigenvalues of the Jacobian, evaluated at the initial point, are -0.067 and -14.9 .

Problem 2 is the system, given by Gear [14],

$$\begin{aligned}x_1' &= -55x_1 + 65x_2 - x_1x_3, & x_1(0) &= 1, \\x_2' &= 0.0785(x_1 - x_2), & x_2(0) &= 1, \\x_3' &= 0.1x_1, & x_3(0) &= 0,\end{aligned}$$

where the eigenvalues at the initial point are $0.0062 \pm 0.01i$ and -55 .

Problem 3 is the elliptic two-body problem, with eccentricity 0.6, given by

$$\begin{aligned}x_1' &= x_3, & x_1(0) &= 0.4, \\x_2' &= x_4, & x_2(0) &= 0, \\x_3' &= -x_1(x_1^2 + x_2^2)^{-3/2}, & x_3(0) &= 0, \\x_4' &= -x_2(x_1^2 + x_2^2)^{-3/2}, & x_4(0) &= 2.\end{aligned}$$

The eigenvalues at the initial point are $\pm \frac{5}{2}\sqrt{5}$ and $\pm \frac{5}{2}\sqrt{\frac{5}{2}}i$.

For these problems a single integration step was carried out using the iteration scheme (3.1) with $\alpha = 1$ and with the modified Newton scheme for comparison. In each case the Jacobian evaluated at $t = 0$ was used and, in the results

presented here, h was set at 0.1 for Problem 1, at $h = 1$ for Problem 2, and at $h = 0.01$ for Problem 3. For both schemes the initial iterate chosen was $Y^0 = x \oplus x \oplus \dots \oplus x$, where x denotes the solution value at the initial point. In practice, extrapolation from previous steps would provide better estimates.

Tests were carried out on several singly implicit methods described by Burrage [2, 3] and Nørsett [16], and also on other methods obtained by applying a transformation given by Hairer and Wanner [15]. The results given here, for three collocation methods, are typical. For each of these methods, the elements of the coefficient matrix $A = [a_{ij}]$ and the elements of the vector $b = [b_i]$ are given by

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad k = 1, 2, \dots, s, \quad i = 1, 2, \dots, s,$$

$$\sum_{i=1}^s b_i c_i^{k-1} = \frac{1}{k}, \quad k = 1, 2, \dots, s,$$

where $\frac{1}{\lambda} c_1, \frac{1}{\lambda} c_2, \dots, \frac{1}{\lambda} c_s$ are the zeros of L_s , the Laguerre polynomial of degree s . The order p is at least s , with $p = s + 1$ if $\frac{1}{\lambda}$ is a zero of the derivative of L_{s+1} .

By Method 1 is meant the collocation method with $s = 2$ and $p = 3$ obtained by choosing λ so that $\frac{1}{\lambda}$ is the largest zero of L'_3 . The method is given by the array

$$\begin{array}{c|cc} \lambda(2 - \sqrt{2}) & \frac{\lambda}{4}(4 - \sqrt{2}) & \frac{\lambda}{4}(4 - 3\sqrt{2}) \\ \lambda(2 + \sqrt{2}) & \frac{\lambda}{4}(4 + 3\sqrt{2}) & \frac{\lambda}{4}(4 + \sqrt{2}) \\ \hline & \frac{1}{2} + \frac{\sqrt{2}}{8} \left(4 - \frac{1}{\lambda}\right) & \frac{1}{2} + \frac{\sqrt{2}}{8} \left(4 - \frac{1}{\lambda}\right) \end{array}, \quad \lambda = \frac{3 + \sqrt{3}}{6}.$$

Method 2 is the collocation method with $s = 3$ and $p = 4$ obtained by choosing $\lambda = \frac{1}{2} + \sqrt{3}/3 \cos \pi/18$, so that $\frac{1}{\lambda}$ is the largest zero of L'_4 .

Method 3 is also a collocation method but in this case $s = 4$ and $p = 4$ with λ chosen so that $\frac{1}{\lambda} = 4.536620297$ is a zero of L_4 . This is one of the choices made by Burrage, Butcher, and Chipman [4].

For the iteration scheme (3.1) with $\alpha = 1$, applied to each method and problem, the quantities

$$e_m = \|Y^m - Y^{m-1}\|, \quad m = 1, 2, 3, \dots,$$

were computed for the uniform norm on \mathbb{R}^N . Minimum values of m , for which $e_m < e$, are given in the table for $e = 5.10^{-4}$, $e = 5.10^{-7}$, and $e = 5.10^{-10}$. For comparison, similar information is given in parentheses for the modified Newton iteration, again with the Jacobian evaluated at the initial point.

The results indicate that both the proposed scheme and the modified Newton iteration perform badly with Method 2. The reasons for this are not known, but

TABLE
 Number of iterates m needed to give $\|Y^m - Y^{m-1}\| < e$.
 Figures for the modified Newton iteration are given in parentheses.

e	Method 1	Method 2	Method 3	
5.10^{-4}	4 (3)	5 (4)	6 (3)	Problem 1 $h = 0.1$
5.10^{-7}	6 (5)	7 (7)	8 (4)	
5.10^{-10}	9 (7)	11 (10)	10 (6)	
5.10^{-4}	4 (3)	6 (3)	6 (3)	Problem 2 $h = 1$
5.10^{-7}	6 (4)	8 (5)	9 (4)	
5.10^{-10}	8 (6)	10 (7)	11 (5)	
5.10^{-4}	4 (3)	6 (3)	5 (3)	Problem 3 $h = 0.01$
5.10^{-7}	5 (4)	8 (4)	8 (3)	
5.10^{-10}	7 (5)	10 (6)	9 (4)	

more extensive tests confirm that the absolute performance of each scheme does depend strongly on the choice of method, a factor that needs to be considered in the design of algorithms. For a given method, the comparative performance of the two schemes may be assessed by estimating an average value for \bar{p} , the relative increase in the number of steps needed by the scheme (3.1) to obtain $e_m < e$. A range of tests was carried out using a variety of problems, including the three given here, with various step lengths and initial values. For $e = 5.10^{-10}$ average values for \bar{p} were obtained of about 0.3 for Method 1, about 0.4 for Method 2, and about 0.9 for Method 3. Similar values were obtained for other choices of e . These values indicate that the scheme (3.1) may be a useful alternative to the modified Newton method for problems of low dimension. The results indicate that competitiveness is lost as the dimension is increased. However, because $\bar{p} \ll s$, the scheme has a marked advantage for parallel computation, and then competitiveness is retained for problems of high dimension.

BIBLIOGRAPHY

1. T. A. Bickart, *An efficient solution process for implicit Runge-Kutta methods*, SIAM J. Numer. Anal. **14** (1977), 1022-1027.
2. K. Burrage, *A special family of Runge-Kutta methods for solving stiff differential equations*, BIT **18** (1978), 22-41.
3. —, *Efficiently implementable algebraically stable Runge-Kutta methods*, SIAM J. Numer. Anal. **19** (1982), 245-258.

4. K. Burrage, J. C. Butcher, and F. H. Chipman, *An implementation of singly-implicit Runge-Kutta methods*, BIT **20** (1980), 326–340.
5. J. C. Butcher, *On the implementation of implicit Runge-Kutta methods*, BIT **16** (1976), 237–240.
6. ———, *Some implementation schemes for implicit Runge-Kutta methods* (Proc. Dundee Conf. on Numerical Analysis 1979), Lecture Notes in Math., vol. 773, Springer-Verlag, pp. 12–24.
7. J. R. Cash, *On a class of implicit Runge-Kutta procedures*, J. Inst. Math. Appl. **19** (1977), 455–470.
8. F. H. Chipman, *The implementation of Runge-Kutta implicit processes*, BIT **13** (1973), 391–393.
9. A. G. Collings and G. J. Tee, *An analysis of Euler and implicit Runge-Kutta numerical integration schemes for structural dynamic problems*, Proc. Sixth Australasian Conf. on the Mechanics of Structures and Materials 1977, vol. 1, pp. 147–154.
10. G. J. Cooper and J. C. Butcher, *An iteration scheme for implicit Runge-Kutta methods*, IMA J. Numer. Anal. **3** (1983), 127–140.
11. H. T. Davis, *Introduction to nonlinear differential and integral equations*, Dover, New York, 1962.
12. W. H. Enright, *Improving the efficiency of matrix operations in the numerical solution of ODEs*, Technical Report no. 98, Computer Science Dept., Univ. of Toronto, 1976.
13. R. Frank and C. W. Ueberhuber, *Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations*, BIT **17** (1977), 146–159.
14. C. W. Gear, *The automatic integration of stiff ordinary differential equations*, Proc. IFIP Congress 1968, pp. 187–193.
15. E. Hairer and G. Wanner, *Algebraically stable and implementable Runge-Kutta methods of high order*, SIAM J. Numer. Anal. **18** (1981), 1098–1108.
16. S. P. Nørsett, *Runge-Kutta methods with a multiple real eigenvalue only*, BIT **16** (1976), 388–393.
17. J. M. Varah, *On the efficient implementation of implicit Runge-Kutta methods*, Math. Comp. **33** (1979), 557–561.

SCHOOL OF MATHEMATICAL AND PHYSICAL SCIENCES, UNIVERSITY OF SUSSEX, BRIGHTON BN1 9QH, ENGLAND