J. Inst. Maths Applics (1979) **24**, 293–301

Diagonally Implicit Runge–Kutta Formulae with Error Estimates

J. R. CASH

Department of Mathematics, Imperial College, South Kensington, London S.W.7

[Received 24 February 1978 and in revised form 4 December 1978]

A class of embedded, diagonally implicit Runge–Kutta formulae suitable for the approximate numerical integration of stiff systems of first order ordinary differential equations is discussed. As well as being computationally efficient, these formulae have the additional facility, that an estimate of the local truncation error is available at every step and this estimate entails virtually no extra computational cost. The formulae derived are compared with various existing methods for a selection of stiff problems and are shown to be superior in certain cases.

1. Introduction

RECENTLY the problem of integrating stiff systems of first order ordinary differential equations of the form

$$y' = f(x, y), \qquad y(x_0) = y_0, \qquad y \in \mathbb{R}^s \tag{1.1}$$

numerically using implicit Runge–Kutta formulae has received a considerable amount of attention. This is due primarily to the fact that it is possible to obtain A-stable implicit Runge–Kutta formulae of the type discussed by Butcher (1964*a*/*b*) of arbitrarily high order (Ehle, 1969) whereas the order of an A-stable linear multistep method is limited to 2, Dahlquist (1963).

Following Alexander (1977) we write our implicit Runge–Kutta method in the form

where

$$\left.\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{q} b_i f(t_n+\tau_i h, y_{n,i}) \\ y_{n,i} &= y_n + h\sum_{j=1}^{q} a_{ij} f(t_n+\tau_j h, y_{n,j}). \end{aligned}\right\} \tag{1.2}$$

Formula (1.2) can be characterized by displaying its coefficients as a Butcher matrix of the form

$$\begin{array}{c|ccc} a_{11} & \cdots & a_{1q} & \tau_1 \\ \vdots & & \vdots & \\ a_{q1} & \cdots & a_{qq} & \tau_q \\ \hline b_1 & \cdots & b_q & \end{array}$$

0020-2932/79/030293+09 $02.00/0

Formula (1.2) is said to be explicit if $a_{ij} = 0$ for $i \leqslant j$, semi-implicit if $a_{ij} = 0$ for $i < j$ and fully implicit otherwise. It is well known that the computational effort involved in using a semi-implicit Runge–Kutta formula is in general considerably less than that which is required by fully implicit Runge–Kutta formulae. A particularly efficient class of semi-implicit *R–K* formulae was first suggested by Nørsett (1974) who considered the case where the a_{ii} are all equal and non-zero. These formulae were further studied by Crouzeix (1975) and by Alexander (1977) and were dubbed Diagonally Implicit Runge–Kutta formulae (DIRK) by Alexander. An additional refinement introduced by Alexander was to derive formulae which are strongly *S*-stable (Prothero & Robinson, 1974) and on at least one of his test problems strongly *S*-stable formulae performed better than ones which were only *A*-stable. The main result proved by Alexander is that an *A*-stable DIRK formula is strongly *S*-stable if it has a Butcher matrix of the form

$$\begin{array}{cccccc|c}
\alpha & & & & & & \tau_1 \\
a_{21} & \alpha & & & & & \tau_2 \\
\vdots & & & & & & \\
a_{q-1,1} & a_{q-1,2} & \cdots & a_{q-1,q-2} & \alpha & & \tau_{q-1} \\
b_1 & b_2 & & b_{q-2} & b_{q-1} & \alpha & 1 \\
\hline
b_1 & b_2 & & b_{q-2} & b_{q-1} & \alpha &
\end{array}$$

where $\alpha > 0$ and where the usual row sum conditions hold. Alexander showed that there are strongly *S*-stable DIRK formulae of order 2 in 2 stages and order 3 in 3 stages but that there does not exist a strongly *S*-stable DIRK formula of order 4 in 4 stages.

The purpose of this paper is to apply Alexander's general results to obtain embedded strongly *S*-stable DIRK formulae. These have a built in estimate of the local truncation error available and, as a result, the steplength of integration may be controlled at virtually no extra computational cost. The idea of embedding formulae to obtain local error estimates is well known and was first proposed by Fehlberg (1964) for use with explicit Runge–Kutta formulae.

In the next section we will give a third order DIRK formula requiring three stages per step and a fourth order one requiring five stages per step both of which have an embedded error estimate. The idea of embedding semi-implicit Runge–Kutta formulae has also been considered by Nørsett (1974). He considered adding an additional stage to a *p*th order formula and this corresponds to embedding a *p*th order formula in one of order $p+1$. The particular formulae which Nørsett considered were for the cases $p = 2, 3$ and the resulting algorithms were *L*-stable but not strongly *S*-stable. In Section 3 we compare the formulae derived in this paper with those considered by Nørsett and Alexander for some stiff test problems.

2. Some Particular Formulae

In this section we derive some strongly *S*-stable DIRK formulae with formulae of lower order embedded. Throughout this section we rely heavily on the results of

Prothero & Robinson (1974), Crouzeix (1975) and Alexander (1977) and the reader is referred to their original papers for the relevant details.

2.1 *Third Order Strongly S-Stable Formula*

This is given by

α	0	0	α
$\tau_2-\alpha$	α	0	τ_2
b_1	b_2	α	1
b_1	b_2	α	

where α is a root of the cubic

$$x^3-3x^2+3x/2-1/6=0,$$

lying in (1/6, 1/2) and

$$\left.\begin{aligned}\tau_2 &= (\alpha^2-3\alpha/2+1/3)/(\alpha^2-2\alpha+1/2)\\ b_1 &= (\tau_2/2-1/6)/((\tau_2-\alpha)(1-\alpha))\\ b_2 &= (\alpha/2-1/6)/((\alpha-\tau_2)(1-\tau_2))\end{aligned}\right\} \tag{2.1}$$

2.2 *Embedded Formula of Order* 2

α	0	α
$\tau_2-\alpha$	α	τ_2
c_1	c_2	

where

$$c_1=(\tau_2-1/2)/(\tau_2-\alpha), \qquad c_2=(\alpha-1/2)/(\alpha-\tau_2). \tag{2.2}$$

The vital point to note about these embedded formulae is that virtually no extra work is required to compute the second order solution once the third order solution has been computed. This is due to the fact that the quantities

$$f(t_n+\tau_i h, y_{n,i}), \quad i=1,2,$$

will already have been computed.

2.3 *Fourth Order Strongly S-Stable Formula*

This has the form

α					τ_1
a_{21}	α				τ_2
a_{31}	a_{32}	α			τ_3
a_{41}	a_{42}	a_{43}	α		τ_4
b_1	b_2	b_3	b_4	α	1
b_1	b_2	b_3	b_4	α	

The necessary and sufficient conditions for this scheme to be of order 4 are given by Alexander in his formulae (2.4.1)–(2.4.4). A particular solution of these equations is

$$\left.\begin{aligned} \alpha &= 0{\cdot}4358665215, & a_{21} &= -1{\cdot}13586652150,\\ a_{31} &= 1{\cdot}08543330679 & a_{32} &= -0{\cdot}721299828287,\\ a_{41} &= 0{\cdot}416349501547, & a_{42} &= 0{\cdot}190984004184,\\ a_{43} &= -0{\cdot}118643265417, & \tau_1 &= \alpha, \quad \tau_2 = -0{\cdot}7,\\ \tau_3 &= 0{\cdot}8, & \tau_4 &= 0{\cdot}924556761814\\ b_1 &= 0{\cdot}896869652944, & b_2 &= 0{\cdot}0182725272734,\\ b_3 &= -0{\cdot}0845900310706, & b_4 &= -0{\cdot}266418670647. \end{aligned}\right\} \tag{2.3}$$

Here the coefficients are given correct to 12 significant figures.

2.3 *Embedded Formula of Order* 3

This takes the form

$$\begin{array}{cccc|c} \alpha & & & & \tau_1\\ a_{21} & \alpha & & & \tau_2\\ a_{31} & a_{32} & \alpha & & \tau_3\\ a_{41} & a_{42} & a_{43} & \alpha & \tau_4\\ \hline c_1 & c_2 & c_3 & c_4 & \end{array}$$

Since this formula is only to be used for the purposes of error estimation there is no need for it to be A-stable. A particular third order formula is given by

$$\left.\begin{aligned} c_1 &= 0{\cdot}776691932910, & c_2 &= 0{\cdot}0297472791484,\\ c_3 &= -0{\cdot}0267440239074, & c_4 &= 0{\cdot}220304811849. \end{aligned}\right\} \tag{2.4}$$

2.4 *Embedded Formula of Order* 2

This takes the form

$$\begin{array}{cc|c} \alpha & & \tau_1\\ a_{21} & \alpha & \tau_2\\ \hline d_1 & d_2 & \end{array}$$

A particular solution is

$$d_1 = (\tau_2 - 1/2)/(\tau_2 - \tau_1), \qquad d_2 = (1/2 - \tau_1)/(\tau_2 - \tau_1). \tag{2.5}$$

It is clear that this approach may be extended in a straightforward manner but we shall not carry out the extension in this paper since the derivation of higher order Runge–Kutta formulae is a long and tedious business and is beyond the scope of this paper. An important point to note is that we have not sacrificed any orders of

accuracy by adopting this approach. This is due to the fact that our third order strongly S-stable formula is based on a 3 stage scheme and our fourth order formula is based on a 5 stage scheme and Alexander has shown that it is not possible to obtain strongly S-stable DIRK formulae of orders 3 or 4 in less than 3 or 5 stages respectively.

We now consider a procedure for estimating the local truncation error of our formulae and for controlling the steplength of integration. We shall in fact describe our algorithm for use with the fourth order scheme (2.3) but the extension to other schemes will be immediate. Suppose that the finally accepted approximation y_n to $y(x_n)$ has been computed and that we wish to compute an approximate solution at $x_{n+1} = x_n + h$. Assuming that y_n is exact and denoting the solution obtained at x_{n+1} using the embedded third order formula by $\hat{y}_{n+1}$ and that obtained at x_{n+1} using the fourth order formula by $\bar{y}_{n+1}$ we have that an estimate $\bar{E}$ to the local truncation error of the asymptotically less accurate solution $\hat{y}_{n+1}$ is

$$\bar{E} = \bar{y}_{n+1} - \hat{y}_{n+1}.$$

If this is less than a prescribed tolerance, in a sense to be defined later, local extrapolation is performed and it is the solution $\bar{y}_{n+1}$ which is actually carried forward. If we assume that a local error tolerance, Tol, is imposed at each step, h and h' are the current step and the next step to be chosen respectively and if we set

$$E = \|\bar{y}_{n+1} - \hat{y}_{n+1}\|_\infty$$

then the steplength is controlled in the following way:

(a) If $E > \text{Tol}$, $h' = h/2$ and start again from x_n.

(b) If $\text{Tol}/\mu < E < \text{Tol}$, $h' = h$. Here $\mu = 2^5 + 2^4$.

(c) If $E < \text{Tol}/\mu$, $h' = 2h$.

Note that this procedure is based on an error per step rather than an error per unit step criterion since the former has been found to be more efficient. Here the factor μ is introduced to make sure that the steplength is not doubled too often when it is not safe to do so. The choice of μ is somewhat arbitrary but practical experience based on third and fourth order formulae has shown that for the cases $k = 3$ and 4 the choice $\mu = 2^k + 2^{k+1}$ is adequate for our purposes.

3. Numerical Results

In this section we present some numerical results obtained using the fourth order formula (2.3) derived in the previous section. In his paper Alexander compared the performance of various semi-implicit Runge–Kutta formulae with that of Gear's method and his general conclusion was that DIRK schemes out-performed Gear's algorithm when low accuracy was required but Gear became generally superior as more strict error tolerances were imposed. The main purpose of this section is to compare the performance of our fourth order strongly S-stable scheme with that of Alexander's third order DIRK scheme and one of Nørsett's embedded formulae. The comparison between our scheme and Gear's algorithm is then immediate through an

examination of Alexander's results although he uses a somewhat different step control procedure from that which we shall use. What we shall in fact show in this section is that, for the test problems which we have chosen, our fourth order scheme with its very cheap error estimate is generally superior to Alexander's scheme in that it requires considerably less Jacobian evaluations for any given error tolerance and generally produces a higher degree of precision. It is fairly obvious that Alexander's scheme with a "two-and-one" error estimate is going to be less efficient than a scheme employing embedding, but nevertheless it is interesting to see exactly how the 2 schemes compare. We also compare the performance of embedded strongly S-stable DIRK schemes with our implementation of Nørsett's schemes for a particular problem and show that the former are generally more efficient. The 3 test problems we consider are:

$$\begin{aligned} y_1' &= 0{\cdot}01-(0{\cdot}01+y_1+y_2)(y_1^2+1001y_1+1001), & y_1(0)&=0\\ y_2' &= 0{\cdot}01-(0{\cdot}01+y_1+y_2)(1+y_2^2), & y_2(0)&=0 \end{aligned} \qquad \text{(P1)}$$

$$\begin{aligned} y_1' &= -y_1+y_2^2+y_3^2+y_4^2, & y_1(0)&=1\\ y_2' &= -10y_2+10(y_3^2+y_4^2), & y_2(0)&=1\\ y_3' &= -40y_3+40y_4^2, & y_3(0)&=1\\ y_4' &= -100y_4+2, & y_4(0)&=1 \end{aligned} \qquad \text{(P2)}$$

$$\begin{aligned} y_1' &= -y_1+2, & y_1(0)&=1\\ y_2' &= -10y_2+20y_1^2, & y_2(0)&=1\\ y_3' &= -40y_3+80(y_1^2+y_2^2), & y_3(0)&=1\\ y_4' &= -100y_4+200(y_1^2+y_2^2+y_3^2), & y_4(0)&=1. \end{aligned} \qquad \text{(P3)}$$

The first system has been suggested as a test problem by Liniger & Willoughby (1969) while the second and third problems are 2 of those suggested by Enright, Hull & Lindberg (1975). Problem 3 exhibits non-linear coupling from smooth to transient components and presented considerable problems to the implicit Runge–Kutta method tested by Enright *et al.*

For the integration of problem P1 an absolute error test was used. Initially the Jacobian matrix was evaluated at the point $(y_1(0), y_2(0))$ and the coefficient matrix of the modified Newton iteration scheme was decomposed into an LU product. If an absolute accuracy of E was required at each step, successive iterates to the $y_{n,i}$ were computed until either two iterates differed by less than $E/100$ or until a maximum number of four iterations had been performed. If the iteration scheme failed to converge sufficiently rapidly when integrating from x_n to x_{n+1} say, the Jacobian matrix was re-evaluated at $(y_1(n), y_2(n))$, (the finally accepted solution values at x_n), and if the iteration scheme still failed to converge in 4 iterations the steplength was halved. When using the fourth order Runge–Kutta scheme the step control procedure described in the previous section was used. In particular if the error estimate was less than the prescribed tolerance, local extrapolation was performed and the solution from the fourth order formula was carried forward. When using Alexander's scheme the step control procedure based on an $h-h/2$ local extrapolation was used. If this estimate was less than a prescribed maximum, local extrapolation was again

TABLE 1

Tol	Fn. evals.	Jac. evals.	Steps	Maximum rel. error
Alexander's scheme				
10^{-2}	1208	82	30	$0{\cdot}966*10^{-4}$
10^{-3}	1399	110	33	$0{\cdot}916*10^{-4}$
10^{-4}	2059	176	43	$0{\cdot}358*10^{-5}$
10^{-5}	3378	298	63	$0{\cdot}288*10^{-6}$
Scheme (2.3)				
10^{-2}	929	25	48	$0{\cdot}215*10^{-3}$
10^{-3}	1093	37	51	$0{\cdot}539*10^{-5}$
10^{-4}	1474	44	70	$0{\cdot}101*10^{-6}$
10^{-5}	1990	64	85	$0{\cdot}603*10^{-8}$

TABLE 2

Tol	Fn. evals.	Jac. evals.	Steps	Maximum rel. error
Alexander's scheme				
10^{-2}	799	24	24	$0{\cdot}696*10^{-2}$
10^{-3}	1297	26	38	$0{\cdot}121*10^{-2}$
10^{-4}	2229	28	66	$0{\cdot}958*10^{-4}$
Scheme (2.3)				
10^{-2}	526	11	27	$0{\cdot}471*10^{-2}$
10^{-3}	1338	12	68	$0{\cdot}547*10^{-3}$
10^{-4}	4265	9	205	$0{\cdot}166*10^{-3}$

TABLE 3

Tol	Fn. evals.	Jac. evals.	Steps	Maximum rel. error
Alexander's scheme				
10^{-2}	1333	78	26	$0{\cdot}770*10^{-2}$
10^{-3}	1954	72	40	$0{\cdot}416*10^{-5}$
10^{-4}	3738	66	80	$0{\cdot}971*10^{-6}$
Scheme (2.3)				
10^{-2}	1577	15	64	$0{\cdot}401*10^{-5}$
10^{-3}	3239	14	132	$0{\cdot}768*10^{-5}$
10^{-4}	7640	16	323	$0{\cdot}842*10^{-6}$

TABLE 4

	Nørsett			(2.1), (2.2)		
Accuracy obtained	Steps	Fns.	Jacs.	Steps	Fns.	Jacs.
$0{\cdot}5*10^{-2}$	24	230	20	17	132	17
$0{\cdot}25*10^{-2}$	35	322	19	23	226	18
$0{\cdot}15*10^{-3}$	98	823	18	34	354	18
$0{\cdot}50*10^{-4}$	116	980	20	68	736	18

performed and it was the solution obtained using a step $h/2$ which was actually carried forward. In Table 1 we give the results obtained for the integration of this problem for a range of error tolerances. Listed are the number of steps, the number of function evaluations and the number of Jacobian evaluations. Also given is the maximum relative error at the point $x = 100$. As can be seen scheme (2.3) produces more accurate results with less computational effort for the range of error tolerances considered. Exactly the same numerical experiment was carried out for problem (P2) except here the range of integration is $0 \leqslant x \leqslant 20$. The results obtained are given in Table 2. As can be seen scheme (2.3) is superior for low orders of accuracy in that it requires relatively few Jacobian evaluations but as stricter error tolerances are imposed Alexander's scheme becomes rather more competitive. In Table 3 we present the results obtained for the integration of problem (P3). Here our numerical experiments were the same as previously described except that a relative rather than an absolute accuracy restriction was imposed. This was needed because of the large values of some of the solution components of the system. As can be seen from Table 3, for problems where Jacobian evaluations and *LU* factorizations are considerably more expensive than other operations, our fourth order scheme is superior due to the relatively few Jacobian evaluations which are required.

Finally in this section we compare the performance, for the integration of (P3) of formulae (2.1) and (2.2) with that of the second order formula used by Nørsett in his program SIRSPN. It is important to note that we are comparing our implementation of these procedures so that they are constrained to attempt the same tasks, i.e. control the relative error in the solution using the step control procedure described in the previous section. Jacobian evaluations were performed whenever iterates did not converge in four or less quasi-Newton cycles and also whenever the steplength was changed. The main purpose of the results presented in Table 4 is to illustrate the effect of local extrapolation. To demonstrate this we give a table of accuracy obtained against the number of steps, function evaluations and Jacobian evaluations. As can be seen, the method employing local extrapolation achieves a given accuracy in less steps and function evaluations than is required by Nørsett's scheme. The number of Jacobian evaluations is about the same for both schemes since they use roughly the same step sequences, and Jacobian re-evaluations are invariably caused by step doubling rather than by the non-convergence of the Newton schemes. The results in Table 4 together with others so far obtained seem to indicate that local extrapolation is advantageous in this context.

References

ALEXANDER, R. 1977 *SIAM J. Numer. Anal.* 1006–1022.
BUTCHER, J. C. 1964*a* *Math. Comput.* **18**, 50–64.
BUTCHER, J. C. 1964*b* *Math. Comput.* **18**, 233–244.
CROUZEIX, M. 1975 Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge–Kutta, Ph.D. thesis, University of Paris.
DAHLQUIST, G. 1963 *BIT.* **3**, 27–43.
EHLE, B. 1969 Ph.D. thesis, University of Waterloo.

ENRIGHT, W., HULL, T. & LINDBERG, B. 1975 *BIT*. **15**, 10–48.
FEHLBERG, E. 1964 *Z. Angew. Math. Mech.* **44**, T17–T29.
LINIGER, W. & WILLOUGHBY, R. A. 1969 Efficient numerical integration methods for stiff systems of differential equations, *IBM Research Report* RC-1970.
NØRSETT, S. P. 1974 Semi-explicit Runge–Kutta methods. *Mathematics and Computation*, No. 6 University of Trondheim.
PROTHERO, A. & ROBINSON, A. 1974 *Math. Comput.* **28**, 145–162.