Journal of Computational and Applied Mathematics 45 (1993) 203–212
North-Holland

CAM 1276

Estimating local truncation errors for Runge–Kutta methods

J.C. Butcher and P.B. Johnston

Department of Mathematics and Statistics, University of Auckland, New Zealand

Received 6 November 1991
Revised 5 February 1992

Abstract

Butcher, J.C. and P.B. Johnston, Estimating local truncation errors for Runge–Kutta methods, Journal of Computational and Applied Mathematics 45 (1993) 203–212.

As an alternative to the use of embedded formulas, it is proposed that local truncation errors might be estimated by a generalization of the method of Ceschino and Kuntzmann (1963). Computed solution values over several successive steps, together with computed derivatives, are used to obtain an accurate approximation to the local truncation error using a Hermite interpolation formula. In this paper, it is shown how a variable-stepsize adaptation of this approximation can be generated cheaply as the solution proceeds. Because the Hermite interpolant will always be available when this procedure is in use, dense output is also available at little additional cost.

Keywords: Initial-value problem; Runge–Kutta methods; local truncation error; Hermite interpolation; variable stepsize; dense output.

1. Introduction

While the most popular methods of local error estimation for Runge–Kutta methods involve the embedding of two or more methods within a single step, the additional cost of this technique is a disadvantage. If further stages are added to allow for the accurate production of dense output, the extra cost, in terms of derivative evaluations, can be considerable and becomes an increasingly heavy overhead as the order increases.

In this paper a method of error estimation proposed originally by Stoller and Morrison [6], then generalized, for the fixed-step case, by Ceschino and Kuntzmann [2] is recalled. More recently, this method has been discussed in [1,5].

The idea is to combine computed y and f values at the end of sufficiently many consecutive steps to enable an error estimate to be computed. If, for example, the stepsize h has been maintained constant over three steps computed using a fourth-order Runge–Kutta method,

Correspondence to: Prof. J.C. Butcher, Department of Mathematics and Statistics, University of Auckland, Auckland, New Zealand.

and the problem being solved is sufficiently smooth, then the solution and derivative values over these steps satisfy

$$\begin{aligned} y(x_{n-3}) &= y_{n-3}, & hy'(x_{n-3}) &= hf(y_{n-3}), \\ y(x_{n-2}) &= y_{n-2} + E + O(h^6), & hy'(x_{n-2}) &= hf(y_{n-2}) + O(h^6), \\ y(x_{n-1}) &= y_{n-1} + 2E + O(h^6), & hy'(x_{n-1}) &= hf(y_{n-1}) + O(h^6), \\ y(x_n) &= y_n + 3E + O(h^6), & hy'(x_n) &= hf(y_n) + O(h^6), \end{aligned} \tag{1.1}$$

where the value of y_{n-3} is regarded as exact and E denotes the principal local truncation error in each of three steps. Hence, by use of Taylor series it is easy to verify that

$$\tfrac{1}{30}(10y_{n-3} + 9y_{n-2} - 18y_{n-1} - y_n) + \tfrac{1}{10}h(f_{n-3} + 6f_{n-2} + 3f_{n-1}) = E + O(h^6) \tag{1.2}$$

and

$$\tfrac{1}{30}(y_{n-3} + 18y_{n-2} - 9y_{n-1} - 10y_n) + \tfrac{1}{10}h(3f_{n-2} + 6f_{n-1} + f_n) = E + O(h^6), \tag{1.3}$$

where $f_{n-k} = f(y_{n-k})$ for $k = 0, 1, 2, 3$. Denote the expressions on the left-hand sides of (1.2) by E_1 and of (1.3) by E_2. Either of these may be used as an estimate of local truncation error.

In Section 2, a generalization of this procedure to the variable-stepsize case is discussed. The aim is to allow stepsizes to vary freely and to compute the variable-stepsize generalization of E_1, or E_2, with the linear combination of solution and derivative values appropriate to the situation, computed as the solution proceeds. To perform this determination of the estimate, with as little extra cost as possible, the use of updatable divided differences is proposed.

As seen above, there is some degree of freedom in the details of the estimation procedure in that either E_1 or E_2 can be used. In Section 3 the use of E_1, E_2, and possible variants, is discussed. In particular, the fixed-stepsize fourth-order case is examined in the hope that the experience gained will give an insight for later studies of the variable-stepsize case.

Section 4 contains the results of some experiments performed with a limited set of test problems. These give sufficient support to the techniques discussed in this paper to warrant further investigations.

In Section 5 a further use of the divided-difference table, generated as an intrinsic part of the error estimation procedure, is discussed as a source of information for producing dense output.

2. The variable-stepsize generalization

To generalize the estimate used in the previous section, suppose that the stepsize in step number n is $h_n = x_n - x_{n-1}$. Then equations (1.1) can be rewritten as follows:

$$\begin{aligned} y(x_{n-3}) &= y_{n-3}, & h_{n-3}y'(x_{n-3}) &= h_{n-3}f(y_{n-3}), \\ y(x_{n-2}) &= y_{n-2} + Er_{n,2}^5 + O(h_n^6), & h_{n-2}y'(x_{n-2}) &= h_{n-2}f(y_{n-2}) + O(h_n^6), \\ y(x_{n-1}) &= y_{n-1} + E(r_{n,1}^5 + r_{n,2}^5) + O(h_n^6), & h_{n-1}y'(x_{n-1}) &= h_{n-1}f(y_{n-1}) + O(h_n^6), \\ y(x_n) &= y_n + E(1 + r_{n,1}^5 + r_{n,2}^5) + O(h_n^6), & h_n y'(x_n) &= h_n f(y_n) + O(h_n^6), \end{aligned} \tag{2.1}$$

Table 1
Confluent divided-difference table

x	Δ_0	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6
x_{n-3}	$Y_{n-3,0}$						
		$Z_{n-3,1}$					
x_{n-3}	$Z_{n-3,0}$		$Y_{n-2,2}$				
		$Y_{n-2,1}$		$Z_{n-2,3}$			
x_{n-2}	$Y_{n-2,0}$		$Z_{n-2,2}$		$Y_{n-1,4}$		
		$Z_{n-2,1}$		$Y_{n-1,3}$		$Z_{n-1,5}$	
x_{n-2}	$Z_{n-2,0}$		$Y_{n-1,2}$		$Z_{n-1,4}$		$Y_{n,6}$
		$Y_{n-1,1}$		$Z_{n-1,3}$		$Y_{n,5}$	
x_{n-1}	$Y_{n-1,0}$		$Z_{n-1,2}$		$Y_{n,4}$		$Z_{n,6}$
		$Z_{n-1,1}$		$Y_{n,3}$		$Z_{n,5}$	
x_{n-1}	$Z_{n-1,0}$		$Y_{n,2}$		$Z_{n,4}$		
		$Y_{n,1}$		$Z_{n,3}$			
x_n	$Y_{n,0}$		$Z_{n,2}$				
		$Z_{n,1}$					
x_n	$Z_{n,0}$						

where $r_{n,1} = h_{n-1}/h_n$, $r_{n,2} = h_{n-2}/h_n$. It is assumed that the stepsize ratios are bounded above and have a positive lower bound.

The next step is to apply the confluent divided-difference operator to $y(x)$ at the points $x = x_n, x_{n-1}, x_{n-1}, x_{n-2}, x_{n-2}, x_{n-3}, x_{n-3}$, to obtain the generalized E_1, and at the points $x = x_n, x_n, x_{n-1}, x_{n-1}, x_{n-2}, x_{n-2}, x_{n-3}$, to obtain the generalized E_2. Assuming that y is sufficiently smooth, then the result computed is $O(h_n^6)$. By applying the divided-difference operations to the terms on the right-hand sides of (2.1), and omitting the $O(h_n^6)$ terms, two confluent divided-difference tables are obtained, one involving computed solutions and computed derivatives and the other involving the coefficients of E occurring in (2.1). Thus approximations to the local truncation error can readily be found.

Once the differences are constructed for the initial fixed steps, it is then necessary only to calculate the lower diagonals of the confluent divided-difference tables as the solution proceeds with each new step. If the fourth-order case is considered, then the ratio of the sixth differences gives the value C, where $E = Ch_n^5$, and hence the variable-stepsize version of the estimate is generated. The two confluent divided-difference tables are initially generated in the format shown in Table 1. In this table, $Y_{m,k}$ and $Z_{m,k}$, where $n-3 \leqslant m \leqslant n$ and $0 \leqslant k \leqslant 6$, are scaled divided differences derived from the usual divided differences $\tilde{Y}_{m,k}$ and $\tilde{Z}_{m,k}$ as follows:

$$Y_{m,k} = h_m^k \tilde{Y}_{m,k} \quad \text{and} \quad Z_{m,k} = h_m^k \tilde{Z}_{m,k}, \tag{2.2}$$

where $\tilde{Y}_{m,0} = \tilde{Z}_{m,0} = y_m$ and $\tilde{Z}_{m,1} = f_m$, so that $\tilde{Y}_{m,1} = (\tilde{Y}_{m,0} - \tilde{Z}_{m-1,0})/h_m$, for the confluent divided-difference table of y. Thus

$$Y_{m,0} = y_m, \qquad Z_{m,0} = y_m, \qquad Y_{m,1} = \frac{Y_{m,0} - Z_{m-1,0}}{S_{m,0}}, \qquad Z_{m,1} = h_m f_m,$$

and for $k \geqslant 2$,

$$Y_{m,k} = \frac{Y_{m,k-1} - r_{m,1}^{-k+1} Z_{m-1,k-1}}{S_{m,a}}, \qquad Z_{m,k} = \frac{Z_{m,k-1} - Y_{m,k-1}}{S_{m,b}}, \tag{2.3}$$

where

$$S_{m,j} = \sum_{i=0}^{j} r_{m,i} \tag{2.4}$$

and

$$a = \lfloor \tfrac{1}{2}(k-1) \rfloor, \qquad b = \lfloor \tfrac{1}{2}(k-2) \rfloor. \tag{2.5}$$

For the confluent divided-difference table of coefficients of E,

$$\tilde{Y}_{m,0} = \tilde{Z}_{m,0} = \begin{cases} \sum_{i=0}^{m-n+2} h_{m-i}^5, & \text{if } n-2 \leqslant m \leqslant n, \\ 0, & \text{if } m = n-3. \end{cases} \tag{2.6}$$

Hence, for $m > n-3$,

$$\tilde{Z}_{m,1} = 0 \quad \text{and} \quad \tilde{Y}_{m,1} = h_m^4.$$

Thus, $Y_{m,0} = Z_{m,0} = \tilde{Y}_{m,0}$, so that $Z_{m,1} = 0$ and $Y_{m,1} = h_m^5$ and the other $Y_{m,k}$, $Z_{m,k}$ follow from (2.3) for $k \geqslant 2$.

If the values of $Y_{n,6}$ and $Z_{n,6}$, in the confluent divided-difference table of y, are denoted by $Y_{n,6}^*$ and $Z_{n,6}^*$, and the corresponding values from the confluent divided-difference table of coefficients of E are denoted by $Y_{n,6}^{**}$ and $Z_{n,6}^{**}$, then, on the assumption that C is constant over the group of steps, the requirement that the sixth differences of the exact solution are approximately constant translates to

$$Y_{n,6}^* + C \times Y_{n,6}^{**} = 0 \quad \text{and} \quad Z_{n,6}^* + C \times Z_{n,6}^{**} = 0. \tag{2.7}$$

Thus the analogous variable-stepsize versions of the estimates E_1 and E_2 are given by

$$E_1 = -\frac{Y_{n,6}^*}{Y_{n,6}^{**}} h_n^5 \quad \text{and} \quad E_2 = -\frac{Z_{n,6}^*}{Z_{n,6}^{**}} h_n^5. \tag{2.8}$$

If the two tables are constructed with an initial three fixed steps, for a fourth-order method, from there on variable steps can be taken and it is only necessary to calculate the two diagonals $Y_{n+1,0} \ldots Y_{n+1,6}$ and $Z_{n+1,0} \ldots Z_{n+1,6}$, in the step from x_n to x_{n+1}, for each difference table. The following algorithm generates the two diagonals of the confluent divided-difference table of y.

Algorithm. *Divided-difference diagonals.*

```
for n := M + 1 to N do
begin
  q[n] := h[n]/h[n − 1];
  S[n, 0] := 1;
  for i := 1 to [(p + 1) div 2] do
    S[n, i] := 1 + S[n − 1, i − 1]/q[n];
  Y[n, 1] := y[n] − y[n − 1];
  Z[n, 1] := h[n] * f[n];
```

```
  for j := 2 to p + 2 do
  begin
    Y[n, j] := (Y[n, j − 1] − q[n]^(j−1) * Z[n − 1, j − 1])/S[n, (j − 1) div 2];
    Z[n, j] := (Z[n, j − 1] − Y[n, j − 1])/S[n, (j − 2) div 2]
  end;
  for i := 0 to [(p + 1) div 2] do S[n − 1, i] := S[n, i];
  y[n − 1] := y[n];
  h[n − 1] := h[n]
end;
```

In the above algorithm, p is the order of the Runge–Kutta method, M is the initial number of fixed steps and N is the total number of steps taken by the solution. The value of M is related to p as follows

$$M = \begin{cases} \frac{1}{2}(p+3), & \text{for } E_1 \text{ when } p \text{ is odd,} \\ \frac{1}{2}(p+1), & \text{for } E_2 \text{ when } p \text{ is odd,} \\ \frac{1}{2}(p+2), & \text{for } E_1 \text{ and } E_2 \text{ when } p \text{ is even.} \end{cases} \tag{2.9}$$

Note that this algorithm is written as though the values of y_n and f_n are available for all steps at the time it is obeyed. In practice, of course, the successive diagonals are computed immediately after the corresponding solution values are evaluated. A slight modification is required to the algorithm for the calculation of the two diagonals of the confluent divided-difference table of coefficients of E. The modification involves changing the values of $Y_{n,1}$ and $Z_{n,1}$ to

```
  Y[n, 1] := h[n]^(p+1);
  Z[n, 1] := 0;
```

and removing the line which updates the value of $y[n-1]$ at the end of the algorithm. It should be noted that the ratio $q[n]$ used in the algorithm is the reciprocal of the ratio $r_{n,1}$.

3. Discussion of various choices

For methods of odd order there are two possible estimates of E, one which does not use the derivative value at x_n and a second which does, but consequently requires the introduction of an extra step. For methods of even order there are also two possible estimates of E, one which does not use the derivative value at x_n and a second which does, both using the same number of steps, e.g., for fourth-order these correspond to E_1 and E_2.

Considering the even-order case, it is noted that the error estimates omit a derivative value from one end of the group of steps. Two further error estimates, which utilize all the derivative values but omit a solution value from one end of the group of steps, can be constructed by taking a suitable linear combination of the first two estimates. Thus, for fourth-order the two estimates are given by $\frac{1}{9}(10E_1 - E_2)$ and $\frac{1}{9}(10E_2 - E_1)$, which are determined as

$$\tfrac{1}{30}(11y_{n-3} + 8y_{n-2} - 19y_{n-1}) + \tfrac{1}{90}h(10f_{n-3} + 57f_{n-2} + 24f_{n-1} - f_n) = E + O(h^6) \tag{3.1}$$

and

$$\tfrac{1}{30}(19y_{n-2} - 8y_{n-1} - 11y_n) + \tfrac{1}{90}h(-f_{n-3} + 24f_{n-2} + 57f_{n-1} + 10f_n) = E + O(h^6). \tag{3.2}$$

Denote the expressions on the left-hand sides of (3.1) and (3.2) by E_3 and E_4. Either of these may also be used as an estimate of local truncation error.

To evaluate the merits of these four estimates of local truncation error, each was applied to problems A1–A5 in the nonstiff DETEST set of [4], using the following two fourth-order methods:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad \text{(Method 1)}, \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{3}{4} & \frac{3}{4} & & & \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} & & \\ 1 & -\frac{1}{3} & -\frac{2}{3} & 2 & \\ \hline & \frac{1}{6} & 0 & \frac{2}{3} & \frac{1}{6} \end{array} \quad \text{(Method 2)}.$$

The algorithm started from $x_0 = 0$ and proceeded in groups of three fixed steps of variable length with the length of the final group of steps reduced to reach the endpoint $x_f = 20$ exactly. The steplength for a group of steps was calculated by the formula

$$h_{\text{new}} = h_{\text{old}} \times \text{SF} \times (\text{Tol}/E)^{1/5}, \tag{3.3}$$

with a safety factor of SF = 0.9 and a tolerance of Tol = 10^{-3}, 10^{-6} and 10^{-9}. The initial steplength was taken as 0.04 in all tests. For each combination of problem, method and tolerance each of the four estimates was used and a comparison made with the exact local truncation error in the sequence of steps. Using an appropriate measure of the overall agreement, the four estimates were placed in order of merit as shown in Table 2.

A reading of Table 2 suggests that there is no clear-cut pattern indicating a uniform advantage for any of the four estimates. It is also seen that the choice between the two

Table 2
Comparison of estimates E_1, E_2, E_3, E_4

Problem	Method	Tol		
		10^{-3}	10^{-6}	10^{-9}
A1	1	E_4, E_2, E_1, E_3	E_4, E_2, E_1, E_3	E_4, E_2, E_1, E_3
	2	E_4, E_2, E_1, E_3	E_4, E_2, E_1, E_3	E_4, E_2, E_1, E_3
A2	1	E_2, E_4, E_1, E_3	E_4, E_2, E_1, E_3	E_2, E_4, E_1, E_3
	2	E_2, E_4, E_1, E_3	E_2, E_4, E_1, E_3	E_4, E_1, E_2, E_3
A3	1	E_1, E_3, E_4, E_2	E_1, E_2, E_4, E_3	E_2, E_3, E_1, E_4
	2	E_2, E_4, E_1, E_3	E_3, E_2, E_4, E_1	E_2, E_1, E_4, E_3
A4	1	E_4, E_2, E_3, E_1	E_1, E_3, E_2, E_4	E_1, E_3, E_2, E_4
	2	E_2, E_4, E_1, E_3	E_1, E_2, E_3, E_4	E_3, E_1, E_2, E_4
A5	1	E_3, E_1, E_2, E_4	E_1, E_3, E_2, E_4	E_3, E_1, E_2, E_4
	2	E_3, E_1, E_2, E_4	E_3, E_1, E_2, E_4	E_3, E_1, E_2, E_4

Table 3
Problem A1 — Global error and number of function evaluations

Tol	Estimate E_1		Estimate E_2		Method 4(5)	
10^{-3}	$-1.27\cdot10^{-5}$	(80)	$-5.17\cdot10^{-5}$	(76)	$-2.36\cdot10^{-4}$	(84)
10^{-6}	$-7.64\cdot10^{-9}$	(180)	$-2.00\cdot10^{-8}$	(172)	$-1.52\cdot10^{-8}$	(198)
10^{-9}	$-1.02\cdot10^{-10}$	(576)	$-1.35\cdot10^{-10}$	(564)	$+2.44\cdot10^{-10}$	(648)

Table 4
Problem A2 — Global error and number of function evaluations

Tol	Estimate E_1		Estimate E_2		Method 4(5)	
10^{-3}	$+1.54\cdot10^{-6}$	(64)	$+3.19\cdot10^{-6}$	(64)	$+1.13\cdot10^{-4}$	(60)
10^{-6}	$-2.07\cdot10^{-8}$	(128)	$-2.17\cdot10^{-8}$	(120)	$+6.67\cdot10^{-7}$	(108)
10^{-9}	$-7.65\cdot10^{-10}$	(332)	$-9.59\cdot10^{-10}$	(312)	$+4.12\cdot10^{-9}$	(306)

underlying Runge–Kutta methods has a significant influence on the preferred estimate. Until more detailed studies show otherwise, there appears to be every reason for choosing either E_1 or E_2, as they are the best estimates in 14 of the 30 tests and the worst estimates in only three of the tests. E_1 and E_2 are also more convenient to incorporate into a program.

4. Experimental results

In this section, the results of some preliminary investigations, aimed at comparing the performance of the estimates proposed here with that of a traditional embedded scheme, are presented.

Table 5
Problem A3 — Global error and number of function evaluations

Tol	Estimate E_1		Estimate E_2		Method 4(5)	
10^{-3}	$+1.17\cdot10^{-2}$	(220)	$+1.55\cdot10^{-2}$	(240)	$-2.44\cdot10^{-2}$	(228)
10^{-6}	$+2.85\cdot10^{-4}$	(604)	$+7.78\cdot10^{-5}$	(616)	$-2.60\cdot10^{-5}$	(690)
10^{-9}	$+1.18\cdot10^{-7}$	(1780)	$+2.53\cdot10^{-7}$	(1776)	$-5.52\cdot10^{-8}$	(2244)

Table 6
Problem A4 — Global error and number of function evaluations

Tol	Estimate E_1		Estimate E_2		Method 4(5)	
10^{-3}	$+2.61\cdot10^{-3}$	(68)	$+2.92\cdot10^{-3}$	(72)	$-9.77\cdot10^{-4}$	(78)
10^{-6}	$+1.33\cdot10^{-5}$	(148)	$+1.48\cdot10^{-5}$	(152)	$-8.64\cdot10^{-6}$	(174)
10^{-9}	$+5.55\cdot10^{-8}$	(520)	$+5.58\cdot10^{-8}$	(520)	$-4.02\cdot10^{-8}$	(570)

Table 7
Problem A5 — Global error and number of function evaluations

Tol	Estimate E_1		Estimate E_2		Method 4(5)	
10^{-3}	$-9.03\cdot10^{-4}$	(60)	$-8.22\cdot10^{-4}$	(64)	$+1.46\cdot10^{-4}$	(66)
10^{-6}	$-5.11\cdot10^{-5}$	(132)	$-1.28\cdot10^{-5}$	(132)	$-3.02\cdot10^{-6}$	(126)
10^{-9}	$-3.59\cdot10^{-7}$	(352)	$-1.05\cdot10^{-7}$	(400)	$-3.57\cdot10^{-8}$	(408)

The variable-stepsize versions of the E_1 and E_2 estimates, in each case applied to the classical fourth-order Runge–Kutta method (Method 1 of Section 3), are compared with a six-stage embedded scheme incorporating a fifth-order method for error estimation purposes, along with a fourth-order method for solution propagation. The latter method, proposed by Fehlberg, is quoted in [1, p.306].

In Tables 3–7, one for each of the DETEST problems A1–A5, three tolerances are used for each of the three local truncation error estimation procedures. The values tabulated are the global error at the endpoint $x_f = 20$, which is reached exactly by the final step, and in parentheses the total number of function evaluations required to reach the endpoint.

A comparison of the results for this small selection of problems, and with the implementations that have been used, suggests that there is not much between the two forms of the new method or between either of them and the embedded method. However, the results are at least encouraging enough to warrant further investigation of the new approach, and work on this is proceeding.

5. Dense output proposal

Gladwell [3] has discussed the use of Hermite interpolation with a Runge–Kutta formula over two successive steps to provide approximations to the solution at points within the steps, i.e., dense output. As a by-product of the error control mechanism described here, dense output is available at a modest cost.

Suppose, for example, that in the implementation of a fourth-order method the generalized version of the E_2 estimate is in use. The confluent divided differences that have been built up for this purpose can be used to provide coefficients in the representation of a fourth-order Hermite interpolation polynomial, written in terms of the basis

$$\left\{1, \frac{x-x_n}{h_n}, \frac{(x-x_n)^2}{h_n^2}, \frac{(x-x_n)^2(x-x_{n-1})}{h_n^3}, \frac{(x-x_n)^2(x-x_{n-1})^2}{h_n^4}\right\}.$$

Although this approach to dense output has some disadvantages over methods which make use of data from the current step, the fact that no additional stages are needed for its facilitation gives it a strong claim to further consideration.

To investigate the quality of the dense output obtained from the divided-difference table the following test was implemented. The variable-stepsize versions of the E_1 and E_2 estimates, in each case applied to the classical fourth-order Runge–Kutta method (Method 1 of Section 3), were used to solve each of the DETEST problems A1–A5. When the solution stepped past an

Table 8
The mean and maximum value of recorded differences for 20 integer points

		Tol					
		10^{-3}		10^{-6}		10^{-9}	
A1	E_1	$1.04\cdot10^{-5}$	$(1.10\cdot10^{-4})$	$6.99\cdot10^{-8}$	$(2.24\cdot10^{-7})$	$1.13\cdot10^{-10}$	$(3.23\cdot10^{-10})$
	E_2	$2.21\cdot10^{-5}$	$(1.73\cdot10^{-4})$	$4.54\cdot10^{-8}$	$(2.17\cdot10^{-7})$	$1.04\cdot10^{-10}$	$(3.16\cdot10^{-10})$
A2	E_1	$9.41\cdot10^{-5}$	$(2.53\cdot10^{-4})$	$9.86\cdot10^{-7}$	$(2.58\cdot10^{-6})$	$5.12\cdot10^{-9}$	$(1.39\cdot10^{-8})$
	E_2	$2.44\cdot10^{-5}$	$(6.60\cdot10^{-5})$	$3.16\cdot10^{-7}$	$(6.21\cdot10^{-7}$	$1.41\cdot10^{-9}$	$(3.55\cdot10^{-9})$
A3	E_1	$2.51\cdot10^{-4}$	$(8.83\cdot10^{-4})$	$1.73\cdot10^{-6}$	$(1.57\cdot10^{-5})$	$2.73\cdot10^{-9}$	$(1.19\cdot10^{-8})$
	E_2	$8.86\cdot10^{-5}$	$(4.85\cdot10^{-4})$	$1.71\cdot10^{-7}$	$(5.46\cdot10^{-7})$	$5.34\cdot10^{-10}$	$(1.83\cdot10^{-9})$
A4	E_1	$5.45\cdot10^{-5}$	$(3.26\cdot10^{-4})$	$6.39\cdot10^{-8}$	$(3.53\cdot10^{-7})$	$5.05\cdot10^{-11}$	$(2.85\cdot10^{-10})$
	E_2	$1.88\cdot10^{-5}$	$(8.16\cdot10^{-5})$	$3.67\cdot10^{-8}$	$(1.70\cdot10^{-7})$	$3.50\cdot10^{-11}$	$(1.57\cdot10^{-10})$
A5	E_1	$6.97\cdot10^{-4}$	$(9.19\cdot10^{-3})$	$8.65\cdot10^{-7}$	$(5.75\cdot10^{-6})$	$2.13\cdot10^{-9}$	$(1.64\cdot10^{-8})$
	E_2	$6.72\cdot10^{-5}$	$(8.46\cdot10^{-4})$	$3.21\cdot10^{-7}$	$(4.04\cdot10^{-6})$	$6.29\cdot10^{-10}$	$(8.36\cdot10^{-9})$

integer point x_j in the interval (0, 20], the algorithm stopped and carried out a backwards interpolation from the endpoint of the step to the integer point to give $y_{i,j}$ for $j = 1, 2, \ldots, 20$. In order to provide a comparison, the algorithm then went back to the beginning of the step and performed a single step out to the integer point to give $y_{m,j}$. The solution stopped at the first step beyond the point $x = 20$.

The difference between the interpolated solution $y_{i,j}$ and the Runge–Kutta solution $y_{m,j}$ should be small, for each j, if the quality of the dense output is good. As a measure of the agreement between these two solutions, the relative difference

$$\frac{|y_{i,j} - y_{m,j}|}{|y_{m,j}|}, \quad \text{for } j = 1, \ldots, 20, \tag{5.1}$$

was recorded at each of the x_j for problems A2–A5. For problem A1 the relative difference was changed to the difference between $y_{i,j}$ and $y_{m,j}$, since the solution to problem A1 is the negative exponential function which approaches zero within the range of the solution and so the use of a relative difference is inappropriate. Hence, for problem A1, $|y_{i,j} - y_{m,j}|$ was recorded. Then the mean and maximum values of recorded differences were calculated for each of the problems using the three tolerances 10^{-3}, 10^{-6} and 10^{-9}. The mean values and, in parentheses, the maximum values are given in Table 8.

It is seen from Table 8 that, for this test, there is good agreement between the interpolated solutions and the Runge–Kutta solutions. In particular, it is noted that the mean value is less than the tolerance used by the Runge–Kutta method in all but a few cases, and in the main this is also true for the maximum value and it is true that all the maximum values are less than 10 times the tolerance.

The test was then modified to incorporate more output points by interpolating the solution at 19 equally-spaced points within each step. The Runge–Kutta solution was calculated at each of these points by taking single steps from the beginning of the step. The mean values and, in

Table 9
The mean and maximum value of recorded differences for 19 interpolation points between steps

		Tol								
		10^{-3}			10^{-6}			10^{-9}		
A1	E_1	$1.21\cdot10^{-5}$	$(2.48\cdot10^{-4})$	779	$4.94\cdot10^{-8}$	$(3.28\cdot10^{-7})$	1615	$1.62\cdot10^{-10}$	$(3.99\cdot10^{-10})$	2679
	E_2	$1.15\cdot10^{-5}$	$(2.35\cdot10^{-4})$	741	$4.97\cdot10^{-8}$	$(3.17\cdot10^{-7})$	1558	$1.64\cdot10^{-10}$	$(3.44\cdot10^{-10})$	2622
A2	E_1	$8.24\cdot10^{-5}$	$(2.85\cdot10^{-4})$	247	$9.00\cdot10^{-7}$	$(2.79\cdot10^{-6})$	551	$4.44\cdot10^{-9}$	$(1.46\cdot10^{-8})$	1520
	E_2	$2.04\cdot10^{-5}$	$(6.63\cdot10^{-5})$	247	$2.36\cdot10^{-7}$	$(6.74\cdot10^{-7})$	513	$1.39\cdot10^{-9}$	$(4.22\cdot10^{-9})$	1425
A3	E_1	$2.83\cdot10^{-4}$	$(1.75\cdot10^{-3})$	760	$1.07\cdot10^{-6}$	$(1.82\cdot10^{-5})$	2242	$2.34\cdot10^{-9}$	$(2.42\cdot10^{-8})$	7733
	E_2	$7.11\cdot10^{-5}$	$(8.93\cdot10^{-4})$	760	$1.96\cdot10^{-7}$	$(1.95\cdot10^{-6})$	2280	$5.18\cdot10^{-10}$	$(8.30\cdot10^{-9})$	7695
A4	E_1	$3.24\cdot10^{-5}$	$(3.43\cdot10^{-4})$	228	$5.91\cdot10^{-8}$	$(5.98\cdot10^{-7})$	646	$5.54\cdot10^{-11}$	$(4.33\cdot10^{-10})$	2413
	E_2	$1.31\cdot10^{-5}$	$(1.11\cdot10^{-4})$	247	$3.64\cdot10^{-8}$	$(2.47\cdot10^{-7})$	646	$4.14\cdot10^{-11}$	$(2.95\cdot10^{-10})$	2413
A5	E_1	$4.77\cdot10^{-4}$	$(6.50\cdot10^{-2})$	228	$3.03\cdot10^{-6}$	$(9.16\cdot10^{-4})$	475	$2.24\cdot10^{-8}$	$(2.72\cdot10^{-5})$	1501
	E_2	$3.50\cdot10^{-5}$	$(1.74\cdot10^{-3})$	247	$2.07\cdot10^{-7}$	$(2.64\cdot10^{-5})$	570	$6.78\cdot10^{-10}$	$(4.18\cdot10^{-7})$	1843

parentheses, the maximum values, together with the total number of interpolation points, are given in Table 9.

Table 9 shows a pattern similar to that in Table 8. It would appear from the results in Tables 8 and 9 that the E_2 estimate is giving better interpolated solutions than the E_1 estimate. This may be explained by the fact that the E_2 estimate is using the nearest derivative values within the group of steps.

References

[1] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods* (Wiley, Chichester, 1986).
[2] F. Ceschino and J. Kuntzmann, *Problèmes Différentiels de Conditions Initiales* (Dunod, Paris, 1963).
[3] I. Gladwell, Initial value routines in the NAG Library, *ACM Trans. Math. Software* **5** (1979) 386–400.
[4] T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, *SIAM J. Numer. Anal.* **9** (1972) 603–637.
[5] J.D. Lawson and B.L. Ehle, Asymptotic error estimation for one-step methods based on quadrature, *Aequationes Math.* **5** (1970) 236–246.
[6] L. Stoller and D. Morrison, A method for the numerical integration of ordinary differential equations, *Math. Comp.* **12** (1958) 269–272.