Highly Continuous Runge–Kutta Interpolants

D. J. HIGHAM
University of Toronto

To augment the discrete Runge–Kutta solution to the initial value problem, piecewise Hermite interpolants have been used to provide a continuous approximation with a continuous first derivative. We show that it is possible to construct interpolants with arbitrarily many continuous derivatives which have the same asymptotic accuracy and basic cost as the Hermite interpolants. We also show that the usual truncation coefficient analysis can be applied to these new interpolants, allowing their accuracy to be examined in more detail. As an illustration, we present some globally C^2 interpolants for use with a popular 4th and 5th order Runge–Kutta pair of Dormand and Prince, and we compare them theoretically and numerically with existing interpolants.

Categories and Subject Descriptors: G 1.7 [**Numerical Analysis**]: Ordinary Differential Equations—*initial value problems, single step methods*; G 4 [**Mathematical Software**]: Algorithm Analysis

General Terms: Performance, Reliability

Additional Key Words and Phrases: Hermite, interpolant, Runge–Kutta

1. INTRODUCTION

In this paper we look at the problem of superimposing a continuous interpolant onto the discretized approximation which arises when an explicit Runge–Kutta (RK) method is used to solve the initial value problem

$$y'(x) = f(x, y(x)), y(a) = y_a \in \mathbf{R}^N, \qquad a \le x \le b. \tag{1.1}$$

Such interpolants are clearly useful for producing dense output and for plotting solution curves. They can also be used to solve root-finding problems such as $g(x, y(x)) = 0$ (see Enright's et al. paper for example [5]) and have been successfully exploited in the integration of problems with low order discontinuities [6]. The availability of an interpolant, $p(x)$, also opens up the possibility of monitoring and controlling the defect (residual) $\delta(x) := p'(x) - f(x, p(x))$, which is a natural measure of the error in the numerical solution [3, 4, 11, 12, 13, 14]. A number of interpolation schemes have been proposed in recent years. These schemes can be assessed according to the three main

Author's address: Department of Mathematics and Computer Science, University of Dundee, DD1 4HN, Scotland.

criteria of cost, accuracy, and degree of global continuity. The cost is usually measured by the number of extra f evaluations per step required to construct the interpolant. The basic measure of accuracy is the asymptotic order of the local error, although more detailed information can be obtained by examining the leading truncation coefficients in the local error expansion. With the exception of some early schemes of Horn [15], most RK interpolants have global C^1 continuity. The main purpose of this work is to show that the standard C^1 Hermite interpolants introduced by Shampine [20] can be adapted so that they have an arbitrary number of continuous derivatives, without affecting either the order of accuracy or the cost. We show how to obtain a local error expansion for the new interpolants, and, for the specific examples that we consider, we quantify the extent to which the local accuracy depends on the mesh distortion.

2. HERMITE INTERPOLANTS

An s-stage, pth order RK formula applied to (1.1) advances the discrete approximation $y_n \approx y(x_n)$ to the point $x_{n+1} = x_n + h_n$ according to

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i k_i^n,$$

where

$$k_i^n = f\left(x_n + c_i h_n, y_n + h_n \sum_{j=1}^{s} a_{ij} k_j^n \right), \qquad i = 1, \ldots, s.$$

The coefficients $\{a_{ij}, b_i, c_i\}_{i,j=1}^{s}$ which define the formula can be displayed in the tableau

$$\begin{array}{c|cccccc|c} c_1 & a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1,s} & b_1 \\ c_2 & a_{21} & & & & & \vdots & b_2 \\ \vdots & \vdots & & & & & \vdots & \vdots \\ \vdots & \vdots & & & & & \vdots & \vdots \\ c_2 & a_{s,1} & \cdots & \cdots & \cdots & \cdots & A_{s,s} & b_s \end{array}$$

We are concerned here with the use of explicit RK formulas, for which $a_{ij} = 0$ when $j \geq i$, although implicit formulas will be needed later for the purpose of analysis. The local solution for the step, $u_n(x)$, is defined by $u_n(x_n) = y_n$ and $u_n'(x) = f(x, u_n(x))$, and the local error satisfies $y_{n+1} - u_n((\mathrm{ix}_{n+1}) = O(h_n^{p+1})$. (Here, and throughout the paper, we assume that f is sufficiently differentiable.) In modern codes, in order to obtain an efficient, reliable solution, the stepsize h_n is varied throughout the integration, with the proviso that some error measure (usually an estimate of the local error in a subsidiary, lower order approximation) is kept within a user-supplied tolerance. An interpolant $p_n(x)$ which approximates $u_n(x)$ over $[x_n, x_{n+1}]$ may also be formed. Typically $p_n(x)$ interpolates to a set of local data which includes y_n, $f(x_n, y_n)$, $y_n + 1$ and $f(x_{n+1}, y_{n+1})$. We say that $p_n(x)$ has *local* *order* $q + 1$ if $p_n(x) - u_n(x) = O(h_n^{q+1})$. A global approximation $p(x)$ may then be constructed by joining together the local interpolants in a piecewise fashion:

$$p(x) := p_n(x) \text{ for } x \in (x_n, x_{n+1}].$$

For a pth order RK formula, the global error $y_{n+1} - y(x_{n+1})$ is $O(h_{\max}^p)$, where $h_{\max} = \max\{h_n\}$, and hence an interpolant of local order p may be sufficient for such purposes as dense output and plotting. However, there are other applications, including some root-finding and defect control algorithms, which require a local order of $p + 1$.

In this section we restrict our attention to the Hermite interpolants which were introduced by Shampine [20]. To define these interpolants, we require the existence of distinct points $\{\xi_i\}_{i=1}^m$ in $[x_n, x_{n+1}]$ with corresponding approximations $u_i \in \mathbf{R}^N$ such that

$$u_i - u_n(\xi_i) = O(h_n^{q+1}), \qquad i = 1, \ldots, m.$$

We also require the derivative data $u_i' := f(\xi_i/, u_i/) \in \mathbf{R}^N$ to be available for $i = 1, \ldots, r \le m$. For Lipschitzian f, this implies that

$$u_i' - u_n'(\xi_i) = O(h_n^{q+1}), \qquad i = 1, \ldots, r.$$

We may take $\xi_1 = x_{\mathrm{n}}$, $u_1 = y_n$, $\xi_r = x_{n+1}$, and $u_r = y_{n+1}$. (Note that $u_r' = f(x_{n+1}, y_{n+1})$ is needed at the start of the next step, and hence is available at no extra cost.) We assume that any additional data u_i and u_i' that is used is generated by adding stages, if necessary, to the RK step. Specifically, we require that the RK tableau can be extended into an explicit t-stage tableau with coefficients $\{a_{ij}, b_i, c_i\}_{i,j=1}^t$ such that $u_i' = k_{\lambda_i}^n$ for some $\lambda_i \in \{1, 2, \ldots, t\}$, and such that each u_i has the form

$$u_i = y_n + h_n \sum_{j=1}^{t} \tilde{b}_{i,j} k_j^n.$$

We then define $p_n(x)$ to be the unique Hermite interpolating polynomial in $\mathbf{P}_{m+r-1}$ which satisfies

$$\begin{aligned} p_n(\xi_i) &= u_i, & i &= 1, \ldots, m, \\ p_n'(\xi_i) &= u_i' & i &= 1, \ldots, r, \end{aligned}$$

where $\mathbf{P}_{m+r-1}$ is the set of polynomials from $\mathbf{R} \to \mathbf{R}^N$ of degree less than or equal to $m + r - 1$.

Shampine examined the accuracy of the above Hermite interpolant and its derivatives, showing that

$$p_n^{(k)}(x) - u_n^{(k)}(x) = O(h_n^{\mathrm{ord}-k}), \qquad 0 \le k \le m + r, \tag{2.1}$$

where ord $= \min\{m + r, q + 1\}$. The result can be interpreted as saying that to achieve a certain order of accuracy, both the order of accuracy of the data, $q + 1$, and the number of pieces of data, $m + r$, must be sufficiently large.

Since adjacent local interpolants $p_n(x)$ and $p_{n+1}(x)$ both interpolate to the solution value y_{n+1} and the derivative value $f(x_{n+1}, y_{n+1})$ at x_{n+1}, it follows that the piecewise polynomial $p(x)$ is continuous and has a continuous first derivative. In the next section, we develop a new class of interpolants which have a higher degree of continuity.

3. FULLY HERMITE INTERPOLANTS

In order to achieve a greater degree of global continuity from a piecewise polynomial interpolant, it is clear that the higher derivatives of the local interpolants must be forced to match at the mesh points. To this end, we consider below the *fully Hermite* interpolant. (Note that there is some ambiguity in the literature over the meaning of the phrase "Hermite interpolant." We use a slight variation of the nomenclature used by Davis [1] in referring to the interpolants defined in Section 2 as Hermite, and the more general interpolants defined below as fully Hermite.)

The fully Hermite interpolating polynomial $q_n(x)$ is defined in terms of the data $\{\mu_i\}_{i=1}^{k}$ and $\{v_i^j\}_{j=0,\, i=1}^{l_i-1,\, k}$ by

$$q_n^{(j)}(\mu_i) = v_i^j, \qquad 0 \le j \le l_i - 1, 1 \le i \le k. \tag{3.1}$$

If the points $\{\mu_i\}_{i=1}^{k}$ in $[x_n, x_{n+1}]$ are distinct, then letting $L = \sum_{i=1}^{k} l_i$, $q_n(x) \in \mathbf{P}_{L-1}$ exists and is unique from the standard interpolation theory [1, p. 28]. We assume that the data has the following asymptotic order of accuracy,

$$v_i^j - u_n^{(j)}(\mu_i) = O\left(h_n^{\hat{q}+1-j}\right). \tag{3.2}$$

The accuracy of $q_n(x)$ can be examined by extending the arguments of Shampine [20]. The analysis is greatly simplified by the introduction of the normalized variable $\tau := (x - x_n)/h_n$. We may write the interpolant in the following form

$$q_n(x_n + \tau h_n) = \sum_{i=1}^{k} \sum_{j=0}^{l_i-1} h_n^j d_{i,j}(\tau) v_i^j, \tag{3.3}$$

where $d_{i,j}(\tau)$ is the unique scalar polynomial of degree not exceeding $L - 1$ which satisfies

$$\begin{aligned} d_{i,j}^{(t)}(\gamma_s) &= 0 \qquad \text{if} \quad i \ne s \text{ or } j \ne t, \qquad \text{for } 0 \le t \le l_i \text{ and } 1 \le s \le k \\ &= 1 \qquad \text{if} \quad i = s \text{ and } j = t \end{aligned}$$

where $\gamma_s = (\mu_s - x_n)/h_n$. The interpolation conditions (3.1) are easily verified. (Note that $d/dx = (1/h_n)d/d\tau$). To examine the local error, we first introduce the polynomial $Q_n(x) \in \mathbf{P}_{L-1}$ which matches the exact data:

$$Q_n^{(j)}(\mu_i) = u_n^{(j)}(\mu_i), \qquad 0 \le j \le l_i - 1, 1 \le i \le k.$$

The local error may then be decomposed into two components

$$q_n(x) - u_n(x) = \left[q_n(x) - Q_n(x)\right] + \left[Q_n(x) - u_n(x)\right],$$

where the first term in square brackets may be thought of as the data error and the second term may be thought of as the interpolation error. Taking equation (3.3) and subtracting the corresponding expression for $Q_n(x_n + \tau h_n)$ we obtain

$$q_n(x_n + \tau h_n) - Q_n(x_n + \tau h_n) = \sum_{i=1}^{k} \sum_{j=0}^{l_i - 1} h_n^j d_{i,j}(\tau)\left[v_i^j - u_n^{(j)}(\mu_i)\right]. \quad (3.4)$$

Since the polynomials $d_{i,j}(\tau)$ are bounded on [0, 1], it follows from (3.2) and (3.4) that the data error satisfies

$$q_n(x) - Q_n(x) = O(h_{nn}^{\hat{q}+1}).$$

Similarly, the data error in the kth derivative satisfies

$$q_n^{(k)}(x) - Q_n^{(k)}(x) = O(h^{\hat{q}+1-k}).$$

The main result of Kansey's paper [17] can be used to give the following expression for the interpolation error,

$$Q_n^{(k)}(x) - u_n^{(k)}(x) = O(h_n^{L-k}), \qquad 0 \le k \le L.$$

Hence, recombining the interpolation and data errors, we find

$$q_n^{(k)}(x) - u_n^{(k)}(x) = O(h_n^{\min(\hat{q}+1,\, L)-k}), \qquad 0 \le k \le L, \quad (3.5)$$

which generalizes (2.1).

To construct an interpolant $q(x)$ with a high degree of global continuity, we must obtain data corresponding to derivatives of $u_n(x)$ at x_n and x_{n+1}. Choosing $\mu_1 = x_n$, $\mu_k = x_{n+1}$, and $l_1 = l_k = D + 1$ will give D continuous derivatives, providing that the data $\{v_1^j\}_{j=0}^{D}$ agrees with the data $\{v_k^j\}_{j=0}^{D}$ from the previous step. One way to obtain such data is to differentiate the Hermite polynomial $p_n(x)$ from Section 2; that is, we set

$$v_k^j = p_n^{(j)}(x_{n+1}), \qquad 0 \le j \le D,$$

and, in order to match the derivative data from the previous step, we set

$$v_1^j = p_{n-1}^{(j)}(x_n), \qquad 0 \le j \le D.$$

For $D \le$ ord, the error result (2.1) shows that

$$v_k^j - u_n^{(j)}(x_{n+1}) = O(h_n^{\text{ord}-j}), \qquad 0 \le j \le D \quad (3.6)$$

and

$$v_1^j - u_{n-1}^{(j)}(x_n) = O(h_{n-1}^{\text{ord}-j}), \qquad 0 \le j \le D. \quad (3.7)$$

(Note that $u_{n-1}(x)$ denotes the local solution for the previous step.) For sufficiently smooth f, we can replace $u_{n-1}(x)$ in (3.7) by the current local solution, to give

$$v_1^j - u_n^{(j)}(x_n) = O(h_n^{\text{ord}-j}), \qquad 0 \le j \le D, \quad (3.8)$$

assuming h_{n-1}/h_n remains bounded above. Hence, with $\hat{q} + 1 = \text{ord}$ in (3.2), using $p_n(x)$ to provide additional pieces of data so that $L \geq \text{ord}$, we see from (3.5) that it is possible to construct an interpolant $q_n(x)$ which is accurate to $O(h_n^{\text{ord}})$. Furthermore, from (3.2), for $j > \text{ord}$, v_1^j and v_k^j do not need *any* asymptotic accuracy, and hence, theoretically, we could achieve an arbitrary degree of continuity $\hat{D}$ by fixing

$$v_1^j = v_k^j = 0, \qquad j = \text{ord} + 1, \dots, \hat{D}.$$

Note that the interpolant $q_n(x)$ defined above has the same asymptotic order of accuracy as the underlying interpolant $p_n(x)$ from which it is defined. Also, the formation of $q_n(x)$ does not require any additional f evaluations (other than those needed to form $p_n(x)$). However, since it is typically a higher degree polynomial, the overhead of evaluating $q_n(x)$ will be greater than that of evaluating $p_n(x)$, and extra storage will be needed for the additional $D - 1$ derivative values which must be passed from step to step. For these reasons, and because of the oscillatory nature of high degree polynomials, the fully Hermite interpolant is only likely to be of practical use when D is reasonably small. Particular examples for which $D = 2$ will be given in the next section.

Like the basic RK method, Hermite interpolants are one-step in nature; that is, $p_n(x)$ does not require any information other than y_n to be carried forward from the previous step. In order to obtain the extra global continuity of the fully Hermite interpolants, however, we were forced to relax the one-step property: $q_n(x)$ is a two-step interpolant since it requires derivatives of $p_{n-1}(x)$ from the previous step. This is reflected by our requirement in (3.8) that h_{n-1}/h_n remains bounded. Intuitively, if the previous stepsize h_{n-1} was much larger than the current stepsize h_n then we would expect the data $p_{n-1}^{(j)}(x_n)$, $j = 2, \dots, D$ to be inappropriate. One of our aims in the next section is to quantify the range of h_{n-1}/h_n for which the data from the previous step is sufficiently accurate.

An alternative approach for constructing a fully Hermite interpolant is as follows: use the derivatives of the previous fully Hermite interpolant at x_n, so that $v_1^0 = y_n$, $v_1^1 = f(x_n, y_n)$, and $v_1^j = q_{n-1}^{(j)}(x_n)$ for $j = 2, 3, \dots, D$, and use only $y_n + 1$ and $f(x_{n+1}, y_{n+1})$ as data at x_{n+1}. The resulting global interpolant will have D continuous derivatives and does not use $p_n(x)$ or $p_{n-1}(x)$ to produce data. In this way it may be possible to avoid computing some of the extra stages needed to form $p_n(x)$. However, with this approach there is a danger that instabilities may arise when the higher derivative data is updated from step to step. The author examined several schemes of this form, with $D = 2, 3, 4, 5$, and found each of them to be unstable; the process of forming $[q_n^{(2)}(x_{n+1}), q_n^{(3)}(x_{n+1}), \dots, q_n^{(D)}(x_{n+1})]$Tinvolves multiplying $[q_{n-1}^{(2)}(x_n), q_{n-1}^{(3)}(x_n), \dots, q_{n-1}^{(D)}(x_n)]$T by a matrix whose spectral radius depends upon the ratio h_{n-1}/h_n and is typically much larger than unity. These instabilities manifested themselves in practice. The Hermite interpolant $p_n(x)$ of Section 2 uses only data computed on the current step and hence will not suffer from such instabilities. Similarly, the fully Hermite interpolant defined earlier in terms of $p_n(x)$ and $p_{n-1}(x)$ will also be stable.

We point out at this stage that it will not always be appropriate to ask for a high degree of continuity in an interpolant, since there are many practical problems where the solution $y(x)$ has a low order discontinuity. It is reasonable to assume that f in (1.1) is continuous, which implies the continuity of $y'(x)$, but in general additional assumptions about f may not be valid. Hence the interpolants presented here are not intended to be used as first-choice interpolants for general purpose codes. However, on problems where it is known that $y(x)$ has many continuous derivatives, the option of a smoother interpolant may be useful. Generally, whenever the interpolant is to be post-processed in some way, the extra smoothness could prove valuable. In particular, there are applications where C^2 continuity is necessary, (see Gladwell et al., [8]). As mentioned by Gladwell et al., [8], a globally C^2 approximation could be obtained by fitting a cubic spline to the data $\{x_n, y_n\}$. However, in this context, cubic splines suffer from the following deficiencies:

(i) With exact data $y_n = y(x_n)$, the error in the cubic spline $C(x)$ has the form

$$\max_{[a,b]} \| C(x) - y(x) \| = O(h_{\max}^4), \qquad h_{\max} = \max\{h_n\},$$

(see, Schultz for example [19, p. 54]). Hence, for a RK formula of order ≥ 5, $C(x)$ cannot match the order of accuracy at the mesh points.

(ii) In general, all the data $\{x_n, y_n\}$ must be computed and stored before the spline can be formed, whereas the fully Hermite interpolant can be generated dynamically and requires only a relatively small amount of information to be passed from step to step.

(iii) Unlike the fully Hermite interpolant, the cubic spline does not satisfy the desirable condition $C'(x_n) = f(x_n, C(x_n))$.

4. TRUNCATION ANALYSIS

The local error in a pth order RK formula can be expanded as

$$y_{n+1} - u_n(x_{n+1}) = h_n^{p+1} \sum_{j=1}^{r_{p+1}} T_j^{(p+1)} F_j^{(p+1)}(x_n, y_n) + O(h_n^{p+2}), \quad (4.1)$$

where the truncation coefficients $T_j^{(p+1)} \in \mathbf{R}$ depend only on the RK formula, and the elementary differentials $F_j^{(p+1)}(x_n, y_n) \in \mathbf{R}^N$ depend only on the differential equations. For simplicity, we write $F_j^{(p+1)}(x_n, y_n) \equiv F_j^{(p+1)}$. For small values of h_n, the local accuracy of y_{n+1} is governed by the leading term in (4.1), and hence it is desirable to make this term as small as possible. Although this term is problem-dependent, it is widely believed that there is some virtue in choosing the RK formula so that some norm of the truncation coefficients is small [2, 21]. (Usually either the Euclidean norm or the infinity norm is used.) Numerical tests have shown that reducing the size of $\| T^{(p+1)} \|$ increases the overall efficiency when a large set of problems is solved, although other constraints on the truncation coefficients should be imposed to ensure that the error estimate behaves reliably, (see Prince and Dormand [18]).

Shampine showed that an analogue of the expansion (4.1) can be obtained for the Hermite interpolants. To see this, first write $p_n(x)$ in normalized form,

$$p_n(x_n + \tau h_n) = \sum_{i=1}^{m} d_{i,0}(\tau) u_i + \sum_{i=1}^{r} h_n d_{i,1}(\tau) u_i'. \tag{4.2}$$

From the form of the data $\{u_i\}_{i=1}^{m}$ and $\{u_i'\}_{i=1}^{r}$, it follows that (4.2) may be re-arranged to give

$$p_n(x_n + \tau h_n) = \sum_{i=1}^{m} d_{i,0}(\tau) y_n + h_n \sum_{i=1}^{t} e_i(\tau) k_i^n, \tag{4.3}$$

where $e_i(\tau)$ is a known polynomial. Using the identity $\sum_{i=1}^{m} d_{i,0}(\tau) \equiv 1$ (which follows from the fact that $g(\tau) \equiv \sum_{i=1}^{m} d_{i,0}(\tau) - 1$ is a polynomial of degree $\leq m + r - 1$ with at least $m + r$ roots, counting multiplicity) we may write

$$p_n(x_n + \tau h_n) = y_n + \tau h_n \sum_{i=1}^{t} \frac{e_i(\tau)}{\tau} k_i^n. \tag{4.4}$$

Thus, for any fixed $\tau \in [0, 1]$ $p_n(x_n + \tau h_n)$ can be regarded as coming from a step of length τh_n with the t-stage explicit RK formula with coefficients $\{a_{ij}/\tau, e_i(\tau)/\tau, c_i/\tau\}_{i,j=1}^{t}$. It follows from (4.1) that the local error can be expanded in the form

$$p_n(x_n + \tau h_n) - u_n(x_n + \tau h_n) = (\tau h_n)^{\text{ord}} \sum_{j=1}^{r_{\text{ord}}} T_j^{(\text{ord})}(\tau) F_j^{(\text{ord})} + O(h_{nn}^{\text{ord}+1}). \tag{4.5}$$

If ord $= p + 1$, then the ratio $\tau^{p+1} \| T^{(p+1)}(\tau) \| / \| T^{(p+1)} \|$ can be used as a basis for assessing the relative local accuracy of the interpolant and the underlying RK formula [9].

At first sight, the style of analysis above does not seem to be applicable to the fully Hermite interpolant $q_n(x)$, since this interpolant makes use of the f evaluations $\{k_i^{n-1}\}_{i=1}^{t}$ from the previous step. However, we show below that $q_n(x_n + \tau h_n)$ can be manipulated into a suitable form by introducing, for the purpose of analysis only, a related $(2t) \times (2t)$ implicit RK tableau.

Denoting the ratio h_{n-1}/h_n by σ, we see that

$$k_i^{n-1} = f\left(x_{n-1} + c_i h_{n-1}, y_{n-1} + h_{n-1} \sum_{j=1}^{t} a_{ij} k_j^{n-1} \right), \qquad i = 1, \ldots, t,$$

may be written

$$k_i^{n-1} = f\left(x_n + (c_i - 1)\sigma h_n, y_n + h_n \left(\sum_{j=1}^{t} a_{ij} \sigma k_j^{n-1} - \sum_{j=1}^{t} b_j \sigma k_j^{n-1} \right) \right), \qquad i = 1, \ldots, t. \tag{4.6}$$

Also, we have

$$v_1^0 := p_{n-1}(x_n) = y_n, \tag{4.7}$$

and, from (4.4),

$$\begin{aligned} v_1^j := p_{n-1}^{(j)}(x_n) &= \left(\frac{1}{h_{n-1}}\right)^j h_{n-1} \sum_{i=1}^{t} e_i^{(j)}(1) k_i^{n-1} \\ &= \left(\frac{1}{\sigma h_n}\right)^{j-1} \sum_{i=1}^{t} e_i^{(j)}(1) k_i^{n-1}, \qquad j = 1, \dots, D. \end{aligned} \tag{4.8}$$

Similarly, for $i = 2, \dots, k$,

$$v_i^0 := p_n(\mu_i) = y_n + h_n \sum_{i=1}^{t} e_i(\gamma_i) k_i^n, \tag{4.8}$$

and

$$v_i^j := p_n^{(j)}(\mu_i) = \left(\frac{1}{h_n}\right)^{j-1} \sum_{i=1}^{t} e_i^{(j)}(\gamma_i) k_i^n, \qquad j = 1, \dots, l_i - 1. \tag{4.9}$$

Substituting (4.7) to (4.9) in (3.3) we find that $q_n(x_n + \tau h_n)$ has the form

$$q_n(x_n + \tau h_n) = \sum_{i=1}^{k} d_{i,0}(\tau) y_n + \tau h_n \sum_{j=1}^{t} \frac{\bar{e}_j(\tau)}{\tau} k_j^n + \tau h_n \sum_{j=1}^{t} \frac{\hat{e}_j(\sigma, \tau)}{\tau} k_j^{n-1}, \tag{4.10}$$

where $\bar{e}_j(\tau)$ is a polynomial in τ, and $\hat{e}_j(\sigma, \tau)$ is a polynomial in τ and a rational polynomial in σ. Hence, using the identity $\sum_{i=1}^{k} d_{i,0}(\tau) \equiv 1$ in (4.10), it follows from (4.6) that $q_n(x_n + \tau h_n)$ may be regarded as the result of a step of length τh_n with the (implicit) $2t$ – stage RK method given by the tableau

$$\begin{array}{c|cc|c} c_1/\tau & & & \bar{e}_1(\tau)/\tau \\ c_2/\tau & & & \bar{e}_2(\tau)/\tau \\ \vdots & \frac{1}{\tau} A & 0 & \vdots \\ c_t/\tau & & & \bar{e}_t(\tau)/\tau \\ (c_1 - 1)\sigma/\tau & & & \hat{e}_1(\sigma, \tau)/\tau \\ (c_2 - 1)\sigma/\tau & & & \hat{e}_2(\sigma, \tau)/\tau \\ \vdots & 0 & \frac{1}{\tau} \hat{A} & \vdots \\ \vdots & & & \vdots \\ (c_t - 1)\sigma/\tau & & & \hat{e}_t(\sigma, \tau)/\tau \end{array} \tag{4.11}$$

where $\hat{a}_{ij} = (a_{ij} - b_j)\sigma$.

Hence, from (4.1), the local error in the interpolant may be written in the form

$$q_n(x_n + \tau h_n) - u_n(x_n + \tau h_n) = (\tau h_n)^{\text{ord}} \sum_{j=1}^{r_{\text{ord}}} \hat{T}_J^{(\text{ord})}(\sigma, \tau) F_J^{(\text{ord})} + O(h_n^{\text{ord}+1}). \tag{4.12}$$

The truncation coefficients $\hat{T}_J^{(\text{ord})}(\sigma, \tau)$ are determined by the tableau (4.11) and are computable for any choices of τ and σ. We may thus gain some insight into the relative accuracy of $q_n(x)$ and $p_n(x)$ by examining the ratio

$$r(\sigma) := \frac{\max_{r \in [0,1]}\{\tau^{\text{ord}} \| \hat{T}^{(\text{ord})}(\tau, \sigma) \|\}}{\max_{\tau \in [0,1]}\{\tau^{\text{ord}} \| T^{(\text{ord})}(\tau) \|\}}. \tag{4.12}$$

Also, by varying σ in (4.12) we can gauge the extent to which the performance of $q_n(x)$ is likely to depend upon the local mesh pattern.

To illustrate the ideas discussed so far, we consider the Dormand–Prince formula pair RK5(4)7FM [2]. We assume that the 5th order formula is used to advance the solution. The method uses 7 stages with the 'first-same-as-last' property ($k_7 = f(x_{n+1}, y_{n+1})$) so that only 6 new function evaluations are required on a general step. Shampine [21] showed that a locally $O(h_n^5)$ approximation $u_2 \approx u_n(x_n + .5h_n)$ can be constructed from $\{k_\iota^n\}_{\iota=1}^7$. Hence with $m = 3$, $r = 2$ and $q + 1 = 5$, a Hermite interpolant with local order 5 is available. We denote this interpolant by $p_n^5(x)$. We may then consider the two fully Hermite interpolants, $q(_n^{5,6}(x)$ and $q_n^{5,7}(x)$, which are defined in terms of $p_n^5(x)$ as follows:

$$\begin{aligned} q_n^{5,6}(x) \colon v_1^J &= p_{n-1}^{5(J)}(x_n), && j = 0,1,2, \\ v_2^J &= p_n^{5(J)}(x_n + 1), && j = 0,1,2, (k = 2, l_1 = l_2 = 3, L = 6) \\ q_n^{5,7}(x) \colon v_1^J &= p_{n-1}^{5(J)}(x_n), && j = 0,1,2, \\ v_2^0 &= p_n^5(x_n + .5h_n), && \\ v_3^j &= p_n^{5(J)}(x_{n+1}), && j = 0,1,2, (k = 3, l_1 = l_3 = 3, l_2 = 1, L = 7). \end{aligned}$$

Both interpolants have local order 5 and possess global C^2 continuity. Note that $q_n^{5,6}(x)$ interpolates to the minimum amount of data, while for $q_n^{5,7}(x)$ we have added the extra value $p_n^5(x_n + .5h_n)$ which is u_2.

A locally 6th order Hermite interpolant for RK5(4)7FM was also given in [21]. Here an extra stage was added to the method and it was shown that an $O(h_n^6)$ approximation $u_2 \approx u_n(x_n + .5h_n)$ could be formed. Hence with $m = r = 3$ and $q + 1 = 6$ the resulting interpolant, $p_n^6(x)$, has local order 6. Based on $p_n^6(x)$ we define the following fully Hermite interpolants,

$$\begin{aligned} q_n^{6,6}(x) \colon v_1^J &= p_{n-1}^{6(J)}(x_n), && j = 0,1,2, \\ v_2^J &= p_6^{6(J)}(x_{n+1}), && j = 0,1,2, (k = 2, l_1 = l_2 = 3, L = 6) \end{aligned}$$

$$
\begin{aligned}
q_n^{6,7}(x):\ & v_1^j = p_{n-1}^{6(j)}(x_n), && j = 0,1,2,\\
& v_2^0 = p_n^6(x_n + .5h_n), && \\
& v_3^j = p_n^{6(j)}(x_{n+1}), && j = 0,1,2, (k = 3, l_1 = l_3 = 3, l_2 = 1, L = 7)\\
q_n^{6,8}(x):\ & v_1^j = p_{n-1}^{6(j)}(x_n), && j = 0,1,2,\\
& v_2^j = p_n^{6(j)}(x_n + .5h_n), && j = 0,1,\\
& v_3^j = p_n^{6(j)}(x_{n+1}), && j = 0,1,2, (k = 3, l_1 = l_3 = 3, l_2 = 2, L = 8).
\end{aligned}
$$

These interpolants have global C^2 continuity and are of local order 6. For $q_n^{6,6}(x)$ we use the minimum 6 pieces of data, and for $q_n^{6,7}(x)$ and $q_n^{6,8}(x)$ we add the extra data u_2 and $\{u_2, u_2'\}$ respectively.

In order to monitor the behavior of $r(\sigma)$ in (4.12) for these new interpolants, it is necessary to determine a reasonable range of values for $\sigma := h_{n-1}/h_n$. Modern codes vary the stepsize in order to take account of the nature of solution, while ensuring that on every step some error test is satisfied. An asymptotic model which predicts the effect on the error estimate of a change in stepsize is used as the basis of the stepsize-changing algorithm. (Precise implementation details of stepsize-changing algorithms for several codes are given by Shampine and Watts [22].) There are two distinct cases to consider:

(i) A successful step from x_{n-1} to $x_{n-1} + h_{n-1} = x_n$ has been taken, and a stepsize $\hat{h}_n$ must be chosen for the next attempted step to $x_n + \hat{h}_n$.

(ii) An attempted step from x_n to $x_n + \hat{h}_n$ failed the error test and a smaller stepsize with which to reattempt the step must be chosen.

For case (i), most codes impose a limit of the form $\hat{h}_n \leq \alpha h_{n-1}$ on the amount by which the stepsize can be increased. It follows that on every step we will have $\sigma \geq 1/\alpha$. A typical value for α is 5. A simple upper bound for σ cannot be derived, because it may be necessary to apply the stepsize reduction (ii) repeatedly before the error test is finally satisfied. Usually codes restrict the factor by which the stepsize may be reduced after a failed step, although a large reduction may be enforced after a fixed number of successive failures. As an example, DVERK [16] uses an asymptotic formula after the first failed step, and then halves the stepsize if successive failures arise. In the testing below, we allow σ to vary between .2 and 8. The case $\sigma = 8$ would arise, for example, if $\hat{h}_n = h_{n-1}$ in (i) and then three stepsize halvings were performed before the step was successful.

In Table I we give values of $r(\sigma)$, measured in the Euclidean norm, for $\sigma = 2^k$, $-1 \leq k \leq 3$. (To obtain the "maximum" values for $\tau \in [0,1]$ in (4.12), we sampled at 20 equally spaced points in [0, 1].) The interpolants $q_n^{5,6}(x)$ and $q_n^{5,7}(x)$ are compared with $p_n^5(x)$, and $q_n^{6,6}(x)$, $q_n^{6,7}(x)$ and $q_n^{6,8}(x)$ are compared with $p_n^6(x)$. Detailed plots of $r(\sigma)$ for σ in the range [.2, 2] are given in Figures 1 -5. (Note that the scale varies between plots.)

Table I. $r(\sigma)$ Values Using the Euclidean Norm

σ	.5	1	2	4	8
$q_n^{5,6}(x)$, $p_n^5(x)$	1.6E0	2.6E0	1.3E1	9.3E1	7.4E2
$q_n^{5,7}(x)$, $p_n^5(x)$	4.9E − 1	4.6E − 1	4.1E0	3.5E1	2.8E2
$q_n^{6,6}(x)$, $p_n^6(x)$	8.1E − 1	1.3E0	1.8E1	2.8E2	4.5E3
$q_n^{6,7}(x)$, $p_n^6(x)$	1.0E0	1.0E0	6.9E0	1.1E2	1.7E3
$q_n^{6,8}(x)$, $p_n^6(x)$	1.0E0	1.0E0	4.0E0	6.1E1	9.7E2

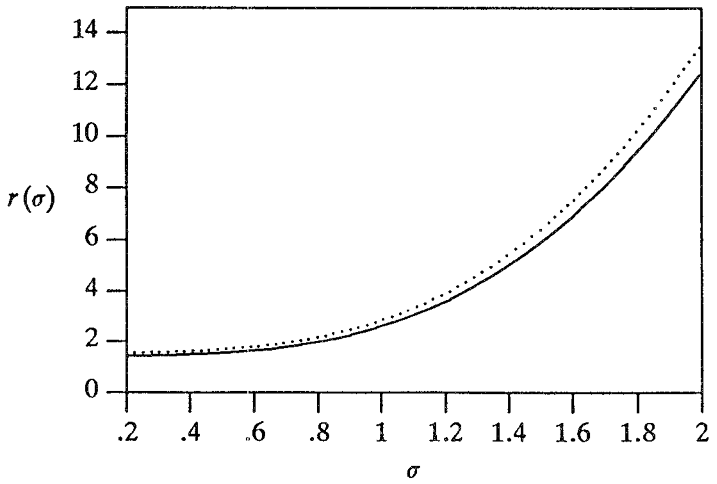

Fig. 1. $r(\sigma)$ values for $q^{5,6}(x)$ and $p^5(x)$ interpolants.
solid line = Euclidean norm
dotted line = infinity norm.

Although a local interpolant is normally used only to provide approximations for $\tau \in [0,1]$, there are aplications where extrapolation to $\tau \in [0,2]$ is necessary; for example, see Enright et al. [6]. Hence, in Table II we present the ratio

$$\tilde{r}(\sigma) := \frac{\max_{\tau\in[0,2]}\{\tau^{\text{ord}}\|\hat{T}^{(\text{ord})}(\tau,\sigma)\|\}}{\max_{\tau\in[0,2]}\{\tau^{\text{ord}}\|T^{(\text{ord})}(\tau)\|\}}.$$

Table I and Figures 1–5, which deal with $\tau \in [0,1]$, indicate that the accuracy of the new interpolants decreases as σ increases beyond 1. For $\sigma = 8$, the truncation coefficients are as much as four thousand times as large as those of the corresponding Hermite interpolant. These results are not surprising, since, as we mentioned in Section 3, if $h_{n-1} \gg h_n$ then the data from the previous step is likely to be less accurate than that obtained during the current step. Further insight is gained by noting that, from (2.1),

$$p_{n-1}^{5(2)}(x_n) - u_{n-1}^{(2)}(x_n) = O(h_{n-1}^3) = O(\sigma^3 h_n^3).$$

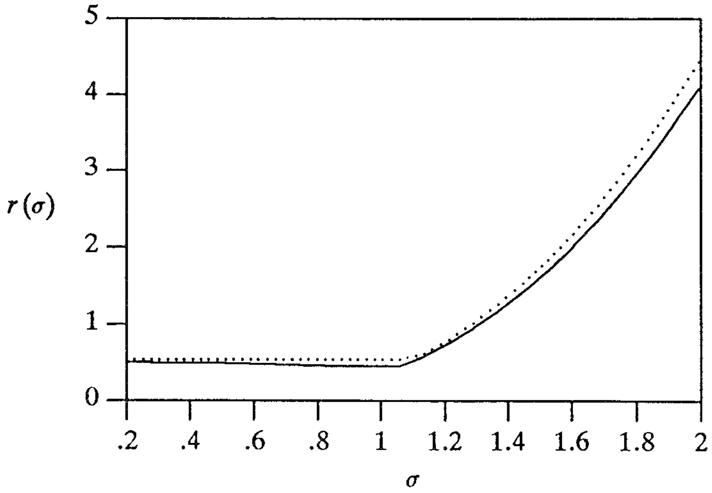

Fig. 2. $r(\sigma)$ values for $q^{5,7}(x)$ and $p^{5}(x)$ interpolants
solid line = Euclidean norm
dotted line = infinity norm.

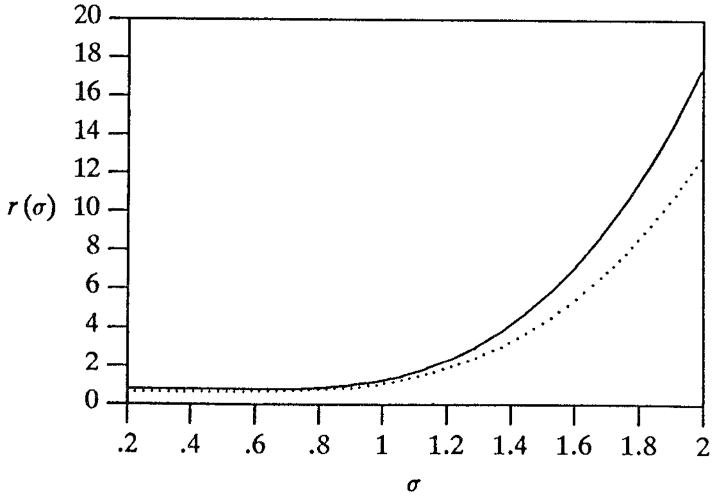

Fig. 3. $r(\sigma)$ values for $q^{6,6}(x)$ and $p^{6}(x)$ interpolants.
solid line = Euclidean norm
dotted line = infinity norm.

So the data error passed to $q_n^{5,6}(x)$ from the previous step should behave like σ^3. Similarly, a σ^4 term arises for the higher order interpolants. Also, we see that in general, for $\sigma > 1$, $q_n^{6,8}(x)$ has smaller $r(\sigma)$ values than $q_n^{6,7}(x)$, which in turn has smaller values than $q_n^{6,6}(x)$. Similarly the $r(\sigma)$ values for $q_n^{5,7}(x)$ are smaller than those for $q_n^{5,6}(x)$. A plausible explanation for this is

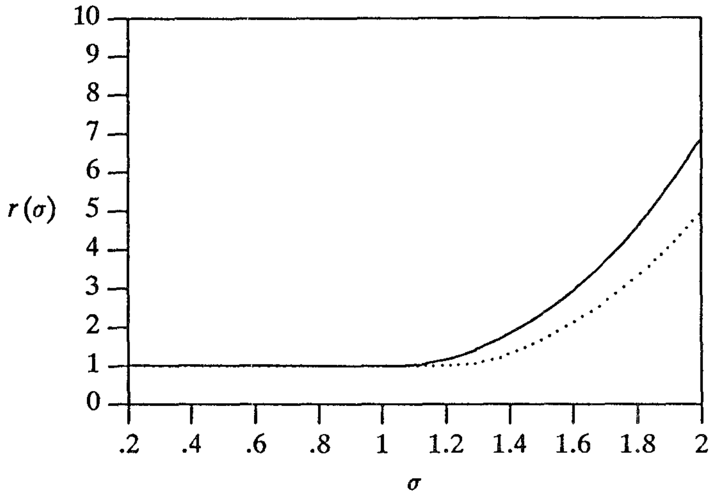

Fig. 4. $r(\sigma)$ values for $q^{6,7}(x)$ and $p^6(x)$ interpolants
solid line = Euclidean norm
dotted line = infinity norm.

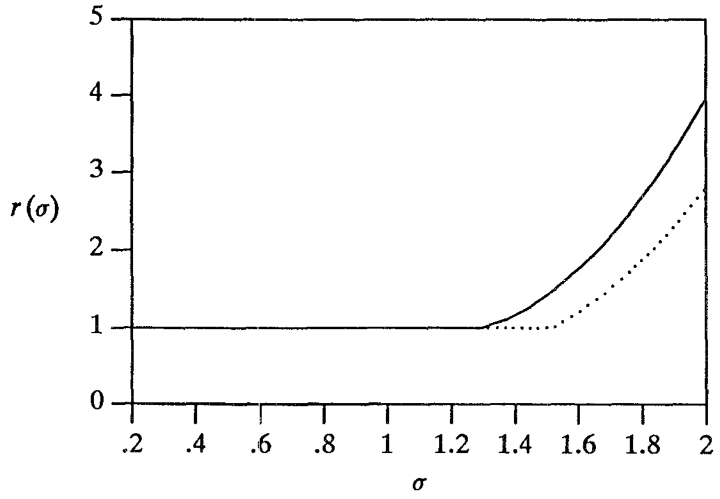

Fig. 5. $r(\sigma)$ values for $q^{6,8}(x)$ and $p^6(x)$ interpolants.
solid line = Euclidean norm
dotted line = infinity norm.

that adding extra data from the current step diminishes the effect of the inaccurate data from the previous step.

Overall, the results suggest that if $\sigma > \approx 2$ with the locally sixth order interpolants, or if $\sigma > \approx 4$ with the locally fifth order interpolants, then the fully Hermites will have much larger local errors than their standard

Table II. $\tilde{r}(\sigma)$ Values using the Euclidean Norm

σ	.5	1	2	4	8
$q_n^{5,6}(x)$, $p_n^5(x)$	6.0E − 1	3.2E − 1	1.9E0	2.0E1	1.6E2
$q_n^{5,7}(x)$, $p_n^5(x)$	2.2E0	3.0E0	9.5E0	6.1E1	4.8E2
$q_n^{6,6}(x)$, $p_n^6(x)$	1.1E0	1.0E0	2.0E0	4 3E1	7.0E2
$q_n^{6,7}(x)$, $p_n^6(x)$	6.9E − 1	1.1E0	8.7E0	1.3E2	2.1E3
$q_n^{6,8}(x)$, $p_n^6(x)$	2.2E0	1.4E0	2.3E1	3.9E2	6.3E3

Hermite counterparts. Note that σ can be computed on every step and hence it would be possible to monitor its size and, if necessary, switch from a C^2 to a C^1 interpolant. (We mention that one of the most common causes of a rapid stepsize reduction is a singularity or low order discontinuity in the local solution. In this case it is obviously inappropriate to ask for too much smoothness from the interpolant.)

5. NUMERICAL TESTING

In this section we record the results of some numerical testing of the new interpolants. For the tests, the RK5(4)7FM pair [2] was implemented with a standard error-control mechanism (locally extrapolated error-per-step) using an absolute local error tolerance of TOL. After a successful step from x_{n-1} to $x_{n-1} + h_{n-1}$, a new stepsize $\hat{h}_n$ for the next attempted step was chosen according to the usual asymptotically-based criterion (see Hall and Higham [10]), subject to the constraint $\hat{h}_n \leq 5h_{n-1}$. After each rejected step, the stepsize was halved and the step retaken.

On every step, each of the interpolants discussed in Section 4 was formed. (On the initial step a fully Hermite interpolant was defined to be the same as the underlying Hermite interpolant.) For each interpolant $r_n(x)$ we found the overall maximum global error based on ten samples from every step, that is,

$$\text{ge}[r] := \max_n \left\{ \max_{1 \leq i \leq 10} \left[\| r_n(x_n + (i/10)h_n) - y(x_n + (i/10)h_n) \|_2 \right] \right\}.$$

Similarly, to monitor the local error we used

$$\text{le}[r] := \max_n \left\{ \max_{1 \leq i \leq 10} \left[\| r_n(x_n + (i/10)h_n) - \hat{u}_n(x_n + (i/10)h_n) \|_2 \right] \right\},$$

where $\hat{u}_n(x_n + (i/10)h_n)$ is the result of a step from x_n to $x_n + (i/10)h_n$ with the 8th order RK formula of the RK8(7)13M pair of Prince and Dormand [18].

We also computed the maximum discontinuity in the second derivatives of the two Hermite interpolants:

$$d^5 := \max_n \{ \| p_n^{5(2)}(x_n) - p_{n-1}^{5(2)}(x_n) \|_2 \}$$

$$d^6 := \max_n \{ \| p_n^{6(2)}(x_n) - p_{n-1}^{6(2)}(x_n) \|_2 \}.$$

Table III. Orbit Problem with $\epsilon = .1$

TOL	10^{-2}	10^{-4}	10^{-6}	10^{-8}
$\text{le}[q^{5,6}]/\text{le}[p^5]$	1.0	1.0	1.1	1.1
$\text{le}[q^{5,7}]/\text{le}[p^5]$	1.0	1.0	1.0	1.0
$\text{le}[q^{6,6}]/\text{le}[p^6]$	1.0	1.0	1.0	1.0
$\text{le}[q^{6,7}]/\text{le}[p^6]$	1.0	1.0	1.0	1.0
$\text{le}[q^{6,8}]/\text{le}[p^6]$	1.0	1.0	1.0	1.0
max. σ	2.0	2.2	1.0	1.0
min. σ	0.5	0.3	0.3	0.3
d^5	2.1E0	2.8E − 2	1.6E − 3	1.1E − 4
d^6	2.7E0	6.3E − 2	8.5E − 4	1.3E − 5

Table IV. Orbit Problem with $\epsilon = .5$

TOL	10^{-2}	10^{-4}	10^{-6}	10^{-8}
$\text{le}[q^{5,6}]/\text{le}[p^5]$	1.0	0.7	2.5	2.5
$\text{le}[q^{5,7}]/\text{le}[p^5]$	1.0	0.8	0.8	0.7
$\text{le}[q^{6,6}]/\text{le}[p^6]$	$1.0^{1\ 0}$	1.0	1.0	1.0
$\text{le}[q^{6,7]/\text{le}[p^6]}$		1.0	1.0	1.0
$\text{le}[q6,8]/\text{le}[p^6]$	1.0	1.0	1.0	1.0
max. σ	4.4	2.0	2.2	1.1
min. σ	0.5	0.5	0.5	0.7
d^5	3.8E0	4.2E − 1	3.3E − 2	2.2E − 3
d^6	6.2E0	2.7E − 1	3.9E − 3	6.3E − 5

Finally, we recorded the maximum and minimum values of σ $(= h_{n-1}/h_n)$. (For the maximum value, we ignored the case where the stepsize on the last step is artificially reduced in order to hit the desired output point b.)

We used the orbit equations [7, Class D]:

$$\begin{aligned}
y_1' &= y_3, & y_1(0) &= 1 - \epsilon, \\
y_2' &= y_4, & y_2(0) &= 0, \\
y_3' &= \frac{-y_1}{\left(y_1^2 + y_2^2\right)^{3/2}} & y_3(0) &= 0, \\
y_4' &= \frac{-y_2}{\left(y_1^2 + y_2^2\right)^{3/2}}, & y_4(0) &= \left(\frac{1+\epsilon}{1-\epsilon}\right)^{1/2}, \qquad 0 \le x \le 20,
\end{aligned}$$

with values of .1, .5, and .9 for the eccentricity parameter ϵ.

The results are presented in Tables III–V. We see that, in terms of the maximum local error,

—$q_n^{6,6}$, $q_n^{6,7}$, $q_n^{6,8}$ and p_n^6 perform almost identically.

—the error in $q_n^{5,6}$ is occasionally smaller than that of p_n^5, but is sometimes larger by a factor of up to 2.6.

—$q_n^{5,7}$ is always at least as accurate as p_n^5.

Table V. Orbit Problem with ϵ = 9

TOL	10^{-2}	10^{-4}	10^{-6}	10^{-8}
le[$q^{5,6}$]/le[p^5]	1.2	0.8	2.6	2.6
le[$q^{5,7}$]/le[p^5]	1.0	0.7	0.7	0.4
le[$q^{6,6}$]/le[p^6]	1.0	1.8	1.0	0.8
le[$q^{6,7}$]/le[p^6]	1.0	1.0	1.0	1.1
le[$q^{6,8}$]/le[p^6]	1.0	1.0	1.0	1.0
max. σ	4.0	2 1	2.2	1.1
min. σ	0.5	0.5	0.5	0.7
d^5	8.7E2	1 2E2	1.0E1	6 1E − 1
d^6	1 2E3	3.8E1	5.8E − 1	1 1E − 2

The corresponding ratios for the maximum *global* errors were 1.0 in all cases. Note that the maximum σ value is typically ≈ 2, and can be as large as 4.4. Hence, given the truncation analysis of Section 4, the competitive performance of the fully Hermite interpolants is perhaps surprising. A more detailed examination of the local errors incurred on every step showed that, on the steps where σ was large ($> \approx 2$), the fully Hermite interpolants had local errors which were greater than those of the corresponding Hermite interpolants by a factor of up to 30. (On such steps the relative performance of the fully Hermite interpolants improved with the amount of extra data taken from the current step, as the results of Section 4 suggest.) However, the local errors on such steps were never as large as the overall maximum values, and hence made no impact on the tabulated results.

The results also show that the Hermite interpolants can have second derivative discontinuities which are several orders of magnitude larger than the local error tolerance. This behavior can be explained as follows. From (2.1), with ord = 5, we have

$$p_{n-1}^{5(2)}(x_n) - u_{n-1}^{(2)}(x_n) = O(h_n^3),$$
$$p_n^{5(2)}(x_n) - u_n^{(2)}(x_n) = O(h_n^3),$$

and similarly, with ord = 6,

$$p_{n-1}^{6(2)}(x_n) - u_{n-1}^{(2)}(x_n) = O(h_n^4),$$
$$p_n^{6(2)}(x_n) - u_n^{(2)}(x_n) = O(h_n^4).$$

Hence, the best results that we can deduce from (2.1) about the second derivative discontinuities are

$$p_n^{5(2)}(x_n) - p_{n-1}^{5(2)}(x_n) = O(h_n^3),$$
$$p_n^{6(2)}(x_n) - p_{n-1}^{6(2)}(x_n) = O(h_n^4).$$

Since the local error estimate, whose norm is being kept smaller than TOL, is an $O(h_n^5)$ quantity, it is reasonable to expect $O(h_n^3)$ and $O(h_n^4)$ quantities to be much larger than TOL.

In summary, the analysis and results presented here show that, on sufficiently smooth problems, unwanted derivative discontinuities can be avoided by the use of fully Hermite interpolants. We have given a theoretical framework for assessing the accuracy of these interpolants, as a function of the local stepsize ratio h_{n-1}/h_n, and we have shown that practical C^2 schemes exist for the Dormand-Prince formula pair RK5(4)7FM.

ACKNOWLEDGMENTS

I thank Wayne Enright, Nick Higham, Ken Jackson and Philip Sharp for commenting on an earlier draft of this manuscript. I also thank an anonymous referee for pointing out the σ^3 and σ^4 behavior of $r(\sigma)$ mentioned in Section 4.

REFERENCES

1. Davis, P. J. *Interpolation and Approximation.* Dover Publications, New York, 1975.
2. Dormand, J. R., and Prince, P. J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* 6 (1980), 19-26.
3. Enright, W. H. A new error-control for initial value solvers. *Appl. Math. Comput. 31* (1989), 288-301.
4. Enright, W. H. Analysis of error control strategies for continuous Runge-Kutta methods. *SIAM J. Numer. Anal. 26* (1989), 588-599.
5. Enright, W. H., Jackson, K. R., Norsett, S. P., and Thomsen, P. G. Interpolants for Runge-Kutta formulas. *ACM Trans. Math. Softw. 12* (1986), 193-218.
6. Enright, W. H., Jackson, K. R., Norsett, S. P., and Thomsen, P. G. Effective solution of discontinuous IVPs using a Runge-Kutta formula pair with interpolants. *Appl. Math. Comput. 27* (1988), 313-335.
7. Enright, W. H., and Pryce, J. D. Two FORTRAN packages for assessing initial value methods. *ACM Trans. Math. Softw. 13* (1987), 1-27.
8. Gladwell, I., Shampine, L. F., Baca, L. S., and Brankin, R. W. Practical aspects of interpolation in Runge-Kutta codes. Numerical Analysis Rep. 102, Univ. of Manchester, (1985).
9. Gladwell, I., Shampine, L. F., Baca, L. S., and Brankin, R. W. Practical aspects of interpolation in Runge-Kutta codes. *SIAM J. Sci. Stat. Comput. 8* (1987), 322-341.
10. Hall, G., and Higham, D. J. Analysis of stepsize selection schemes for Runge-Kutta codes. *IMA J. Numer. Anal. 8* (1988), 305-310.
11. Hanson, P. M., and Enright, W. H. Controlling the defect in existing variable-order Adams codes for initial-value problems. *ACM Trans. Math. Softw. 9* (1983), 71-97.
12. Higham, D. J. Robust defect control with Runge-Kutta schemes, Numerical Analysis Rep. 150, Univ. of Manchester, 1987, To appear in SIAM J. Numer. Anal.
13. Higham, D. J. Defect estimation in Adams PECE codes. *SIAM J. Sci. Stat. Comput. 10* (1989), 964-976.
14. Higham, D. J. Runge-Kutta defect control using Hermite-Birkhoff interpolation. Tech. Rep. 221/89, Dept. of Computer Science, Univ. of Toronto, 1989. To appear in *SIAM J. Sci. Stat. Comput.*
15. Horn, M. K. Fourth- and fifth-order, scaled Runge-Kutta algorithms for treating dense output. *SIAM J. Numer. Anal. 20* (1983), 558-568.
16. Hull, T. E., Enright, W. H., and Jackson, K. R. User's guide for DVERK-a subroutine

for solving non-stiff ODEs. Tech. Rep. 100, Dept. of Computer Science, Univ. of Toronto, 1976.

17. KANSY, K. Elementare Fehlerdarstell ıng fur Ableitungen bei der Hermite-Interpolation. *Numer. Math. 21*, (1973), 350-354.
18. PRINCE, P. J., AND DORMAND, J. R. High order embedded Runge-Kutta formulae. *J. Comput. Appl. Maths. 7* (1981), 67-75.
19. SCHULTZ, M. H. *Spline Analysis.* Prentice Hall, Englewood Cliffs, N.J , 1973.
20. SHAMPINE, L. F. Interpolation for Runge-Kutta methods. *SIAM J. Numer. Anal. 22* (1985), 1014-1027.
21. SHAMPINE, L. F. Some practical Runge-Kutta formulas. *Math. Comp. 46* (1986), 135-150.
22. SHAMPINE, L F., AND WATTS, H. A. The art of writing a Runge-Kutta code II *Appl. Math. Comput. 5* (1979), 93-121.

Received May 1989; revised November 1989; accepted December 1989