SIAM J. SCI. COMPUT.
Vol. 21, No. 1, pp. 227–254

SUPERCONVERGENT INTERPOLANTS FOR THE COLLOCATION SOLUTION OF BOUNDARY VALUE ORDINARY DIFFERENTIAL EQUATIONS*

W. H. ENRIGHT† AND P. H. MUIR‡

Abstract. A long-standing open question associated with the use of collocation methods for boundary value ordinary differential equations is concerned with the development of a high order continuous solution approximation to augment the high order discrete solution approximation, obtained at the mesh points which subdivide the problem interval. It is well known that the use of collocation at Gauss points leads to solution approximations at the mesh points for which the global error is $O(h^{2k})$, where k is the number of collocation points used per subinterval and h is the subinterval size. This discrete solution is said to be superconvergent. The collocation solution also yields a C^0 continuous solution approximation that has a global error of $O(h^{k+1})$. In this paper, we show how to efficiently augment the superconvergent discrete collocation solution to obtain C^1 continuous "superconvergent" interpolants whose global errors are $O(h^{2k})$. The key ideas are to use the theoretical framework of continuous Runge–Kutta schemes and to augment the collocation solution with inexpensive monoimplicit Runge–Kutta stages. Specific schemes are derived for k = 1, 2, 3, and 4. Numerical results are provided to support the theoretical analysis.

Key words. collocation, Runge–Kutta methods, boundary value ordinary differential equations, interpolation

AMS subject classifications. 65L05, 65L10

PII. S1064827597329114

1. Introduction.

The method of collocation at Gauss points has enjoyed considerable success in the numerical solution of boundary value ordinary differential equations (BVODEs). The widely used COLSYS/COLNEW collocation codes [1], [3] employ, respectively, a B-spline basis and a monomial spline basis to provide a piecewise polynomial approximation to the solution of a BVODE, which for the purposes of this paper we will assume has the form

$$y'(t) = f(t, y(t)), \quad t \in [a, b] \qquad y : \Re \to \Re^n, \qquad f : \Re \times \Re^n \to \Re^n, \tag{1}$$

with boundary conditions

$$g(y(a), y(b)) = 0, \quad g : \Re^n \times \Re^n \to \Re^n \tag{2}$$

(although the particular form of the boundary conditions will not be important for this paper). We will assume appropriate conditions on f and g to guarantee the existence and local uniqueness of solutions to the BVODE (see [2, Section 3.1.2]). The above form is referred to as a first order system form. See [2] for a discussion on conversion of more general BVODEs to this form. While many software packages for the numerical solution of BVODEs require the system of equations to be written in the form (1), the COLSYS/COLNEW codes have the advantage in that they can be used to treat mixed order systems directly.

*Received by the editors October 21, 1997; accepted for publication (in revised form) by S. Campbell August 27, 1998; published electronically September 1, 1999.
http://www.siam.org/journals/sisc/21-1/32911.html

†Department of Computer Science, University of Toronto, Toronto, ON, M5S 1A4 Canada, (enright@cs.toronto.edu).

‡Department of Mathematics and Computing Science, Saint Mary's University, Halifax, NS, B3H 3C3 Canada (muir@stmarys.ca). The work of this author was supported by the Natural Sciences and Engineering Research Council of Canada.

The collocation method assumes that the problem domain is partitioned into N subintervals by a mesh, $\{t_i\}_{i=0}^N$, with $t_0 = a$ and $t_N = b$. Letting k be the number of collocation points per subinterval, the continuous collocation solution, on the ith subinterval, $[t_{i-1}, t_i]$, is a polynomial of degree k which can be written in the form

$$\hat{u}_i(t) = \sum_{j=0}^{k} c_{ij} B_{ij}(t), \tag{3}$$

where each $B_{ij}(t)$ is a polynomial basis function of degree k and each $c_{ij} \in \Re^n$ is a vector of unknown coefficients to be determined by requiring $\hat{u}_i(t)$ to satisfy (1) at the collocation points and to satisfy continuity conditions at the interior mesh points, or the boundary conditions, as appropriate. The collocation points are $\widehat{t_{ir}} = t_{i-1} + \rho_r h$, where $h = t_{i-1} - t_i$ and, in COLSYS/COLNEW, the $\{\rho_r\}_{r=1}^k$ are the Gauss points on [0,1].

From the theory for collocation at Gauss points (see, e.g., [4]) it is known that the global error at the mesh points is

$$| \hat{u}_i(t_i) - y(t_i) | \sim O(h^{2k}), \quad i = 0, \dots, N, \tag{4}$$

while the global error at intermediate points is

$$| \hat{u}_i(t) - y(t) | \sim O(h^{k+1}), \quad t \in (t_{i-1}, t_i), \quad i = 0, \dots, N.$$

While the presence of a higher order solution approximation at the mesh points is certainly a positive feature of collocation methods, it has long been recognized that it is inconvenient that the associated collocation polynomial is of substantially lower order. In [20], the author considers augmenting the high order discrete collocation solution by constructing, for each subinterval, a high order, "superconvergent" polynomial interpolant having a global error also of $O(h^{2k})$ and based on mesh point solution approximations from several adjacent subintervals. The author reports some difficulties, one of which is that the numerical stability of the approach depends on the ratio of subinterval sizes; these ratios can be large for the nonuniform meshes which arise in the solution of difficult BVODEs.

In [21] the authors consider constructing a local superconvergent polynomial interpolant having a global error of $O(h^{2k})$, using only information from the current subinterval. On each subinterval, in addition to the (high order) end point solution values, sufficient extra high order derivative information is obtained by performing a "secondary collocation." A Hermite–Birkhoff interpolant is then constructed based on these high order values. While this is perhaps the natural approach to providing a high order local interpolant, there is some question, nonetheless, about the efficiency of this approach since it requires, in general, the solution of a nonlinear system of size $O(nk)$ on each subinterval. Assuming a Newton-type iteration is used to solve these nonlinear systems, with q iterations to convergence, this leads to an overall cost of $O(Nqn^3k^3)$, which is approximately the same operation count as that associated with computing a continuous collocation solution having a global error of $O(h^{2k})$ simply by starting over using $2k$ collocation points per subinterval.

In this paper, we place the question of developing a superconvergent polynomial interpolant in the context of continuous Runge–Kutta (CRK) schemes, which are naturally able to yield high order interpolants using solution and derivative information of various orders. The general form of an efficient CRK scheme, for the ith subinterval, is (see [19] and references within)

$$u_i(t_{i-1} + \theta h) = y_{i-1} + h \sum_{r=1}^{s} b_r(\theta) f(\widehat{t}_r, \widehat{y}_r), \tag{5}$$

where y_{i-1} is an approximation to $y(t_{i-1})$, $b_r(\theta)$ is a polynomial in $\theta \in [0, 1]$, $\widehat{t}_r = t_{i-1} + c_r h$, and

$$\widehat{y}_r = y_{i-1} + h \sum_{j=1}^{s} a_{rj} f(\widehat{t}_j, \widehat{y}_j), \quad r = 1, \ldots, s. \tag{6}$$

The evaluations of the derivatives, $f(\widehat{t}_r, \widehat{y}_r)$, will be referred to as implicit Runge–Kutta (IRK) stages. It is also possible to write (6) in an equivalent "parameterized" form [18]

$$\widehat{y}_r = (1 - v_r) y_{i-1} + v_r y_i + h \sum_{j=1}^{s} x_{rj} f(\widehat{t}_j, \widehat{y}_j), \quad r = 1, \ldots, s, \tag{7}$$

and we will refer to stages written in this form as parameterized IRK (PIRK) stages. In the context of BVODEs it is common (see, e.g., [8], [9], [10], [11], [13], [14], [16]) to employ monoimplicit Runge–Kutta (MIRK) stages, which follow from (7) by requiring the matrix X (whose elements are the x_{rj}'s) to be strictly lower triangular, and thus (7) takes the form

$$\widehat{y}_r = (1 - v_r) y_{i-1} + v_r y_i + h \sum_{j=1}^{r-1} x_{rj} f(\widehat{t}_j, \widehat{y}_j), \quad r = 1, \ldots, s. \tag{8}$$

Note that when both y_{i-1} and y_i are available, the MIRK stages can be computed quite efficiently since the calculations are explicit. Note also that any stage written in MIRK form (8) can be transformed to IRK form (6) (see [18]), while any IRK stage (6) can be trivially written in the form (7) with $v_r = 0$ and $x_{rj} = a_{rj}$.

After an approximate solution is computed by the collocation solver, the discrete solution values at the mesh points and the derivative evaluations at the collocation points will be available. The CRK schemes upon which our superconvergent interpolants are based are then constructed as follows. For each subinterval, the CRK scheme will employ the discrete collocation solution value at the left end point, the derivative evaluations at the collocation points (collocation stages), the derivative evaluations at the end points (end point stages), and sufficient additional MIRK stages (8) to obtain a scheme of the desired order of accuracy. Since the extra MIRK stages and the end point stages can be computed explicitly, and the end point values and collocation stages are already available, the associated computational costs will be low. The idea is a very natural extension of the embedding of a discrete MIRK scheme within a CMIRK scheme, as employed in MIRKDC [14] and as further considered in [17]. The inclusion of the end point and collocation stages within the CRK scheme will define the coefficients for most of the stages; the coefficients for the additional MIRK stages and all the weight polynomials will be determined using the same general approach as in [19], which involves the direct solution of continuous forms of the Runge–Kutta order conditions.

It should be emphasized that the computations for the setup of the superconvergent interpolant occur after the collocation solution has been computed. Thus the stability of the calculation of the global solution is not affected. It should also be noted that the dominant computational cost will be associated with the initial computation of the collocation solution, rather than with the subsequent construction of the superconvergent interpolant.

2. Derivation of interpolants. It is possible to consider the question of deriving or constructing superconvergent interpolants based on CRK schemes from at least two perspectives.

One approach, which we call the Runge–Kutta approach, involves deriving CRK schemes by directly solving certain sets of equations called order conditions which depend on the coefficients of the scheme and characterize all schemes of a given global order of convergence. The goal is to choose the minimum number of stages such that there will be a sufficient number of free parameters to allow the order conditions to be satisfied. An advantage of this approach is its generality; any scheme of the desired order can be obtained in this way provided one is able to choose the coefficients of the scheme to satisfy the order conditions. A disadvantage is that the process of solving the order conditions while using as few free parameters as possible is not straightforward and heuristics must frequently be used to help in the derivation. It may also be necessary to impose extra conditions on the CRK scheme to ensure that it yields an interpolant with C^1 continuity.

The second approach, which we call the boot strap approach, involves deriving interpolants based on CRK schemes within the context of a "boot-strapping" algorithm [12] which we now briefly describe for the current context. As indicated previously, we assume that the discrete superconvergent collocation solution approximations at the mesh points (see (4)) and the lower order collocation stages are available. As well, we assume that the evaluations of the corresponding end point stages have already been performed. The boot-strapping algorithm begins with this information and produces a sequence of interpolants, each one order higher than the previous, until an interpolant of the desired order is obtained. The key idea is that new stages of the appropriate order required for each new interpolant are obtained by using the high order end point information plus evaluations of the previous interpolant, which is one order lower. We will describe the first few steps of this process.

On each subinterval, the natural interpolant associated with the collocation method is the collocation polynomial (3), which is of degree k and has *local error* $O(h^{k+1})$. (On the ith subinterval, of length h, a CRK scheme (5) has a local error of $O(h^{p+1})$ if $|u_i(t) - y_i(t)| \sim O(h^{p+1})$ for $t_{i-1} \leq t \leq t_i$, where $y_i(t)$ is the true solution to the local initial value ODE

$$y'(t) = f(t, y(t)), \quad y(t_{i-1}) = y_{i-1}, \tag{9}$$

where y_{i-1} is the approximate solution value at t_{i-1}.) Here we will refer to the collocation polynomial as $u^{(0)}(t)$. The next interpolant, which we will refer to as $u^{(1)}(t)$, is a polynomial of degree $k+1$ having local error $O(h^{k+2})$ on each subinterval; it is obtained by requiring it to interpolate the end point solution values and by requiring its derivative to interpolate the end point stages, as well as $k-2$ additional stages, $f(t_j^{(1)}, u^{(0)}(t_j^{(1)}))$, $j = 1, \ldots, k-2$. These latter stages have a local error of only $O(h^{k+1})$ but since each is multiplied by h when it appears in the CRK scheme, the contribution to the local error of $u^{(1)}(t)$ is $O(h^{k+2})$. Thus $u^{(1)}(t)$ is uniquely determined by $k+2$ interpolation requirements involving information having a local

TABLE 1
Number of extra stages required to compute interpolant.

k	Boot strap	Boot strap	Runge–Kutta	Runge–Kutta
l.e.	$O(h^{2k})$	$O(h^{2k+1})$	$O(h^{2k})$	$O(h^{2k+1})$
1	0	0	0	0
2	0	1	0	1
3	3	6	1	3
4	9	14	3	8
5	18	25	18	25
6	30	39	30	39
7	45	56	45	56

error of at least $O(h^{k+2})$. The next interpolant, $u^{(2)}(t)$, a polynomial of degree $k+2$, having local error $O(h^{k+3})$, is defined by interpolation conditions based on the end point values and end point stages plus $k-1$ additional stages, having a local error of $O(h^{k+2})$ of the form $f(t_j^{(2)}, u^{(1)}(t_j^{(2)}))$. When multiplied by h, these latter stages, plus the end point information, provide interpolation conditions having a local error of at least $O(h^{k+3})$ and uniquely determine $u^{(2)}(t)$ with a local error of $O(h^{k+3})$. This process continues until an interpolant having the desired local error is obtained.

The final interpolant, associated with each subinterval, includes explicit dependence on the solution and derivative information at both end points of the subinterval and is constructed so that it interpolates these solution values and its derivative interpolates the derivative values. Hence the resultant global interpolant has C^1 continuity.

We conclude this section with a comparison of these two approaches. In Table 1, for $k = 1, \ldots, 7$ (the range of k values available in the COLSYS/COLNEW code), we give the number of *extra* stages that will need to be computed in order to obtain interpolants having local errors of $O(h^{2k})$ and $O(h^{2k+1})$. We will assume that the end point stages and collocation stages are already available. The boot strap data is obtained from a consideration of the boot strap derivation process described above. For $k = 1, \ldots, 4$, the Runge–Kutta data corresponds to the smallest number we have been able to achieve and results justifying these claims are presented in subsequent sections of this paper. For $k = 5, \ldots, 7$, the Runge–Kutta data is equal to that of the boot strap approach; this follows from the observation that the interpolant obtained from the boot strap approach can be rewritten as a CRK scheme. In fact the boot strap approach can be viewed as a strategy for introducing simplifying assumptions to allow the straightforward solution of the order conditions.

For general k, the boot-strapping approach requires $\frac{3k}{2}(k-3)+3$ extra stages to obtain the interpolant having local error $O(h^{2k})$ and $\frac{k}{2}(3k-5)$ extra stages to obtain the interpolant having local error $O(h^{2k+1})$. As observed above, these values also represent upper bounds for the Runge–Kutta approach. Furthermore, we observe from Table 1 that for $k = 1, \ldots, 4$ it is frequently possible, by employing the more general Runge–Kutta approach, to use substantially fewer stages than required by the boot strap approach. Because of this, in the remainder of this paper we will restrict our attention to the Runge–Kutta approach for the derivation of superconvergent interpolants.

3. Background. In this section we describe our approach for deriving C^1 superconvergent interpolants for discrete collocation solutions based on the use of CRK schemes.

When the CRK scheme (5) uses PIRK stages (7), the coefficients are usually written in a tableau,

$$\begin{array}{c|c|c} c & v & X \\ \hline & & b(\theta)^T \end{array},$$

where $c = [c_1, \ldots, c_s]^T$, $v = [v_1, \ldots, v_s]^T$, $b = [b_1(\theta), \ldots, b_s(\theta)]^T$, and the i, jth component of the s by s matrix X is x_{ij}. A CRK scheme is of *order* p if its local error, on a subinterval of length h, is $O(h^{p+1})$. The rth weight polynomial, $b_r(\theta)$, is a polynomial in θ of degree p. A CRK scheme of order p is determined by requiring its coefficients and weight polynomials to satisfy certain equations, called continuous order conditions. Another important attribute of a CRK scheme is its stage order. A CRK scheme with PIRK stages has stage order $q(\leq p)$ if it satisfies (see [5])

$$SO_l \equiv Xc^{l-1} + \frac{v}{l} - \frac{c^l}{l} = 0, \quad l = 1, \ldots, q, \tag{10}$$

where $c^l = [c_1^l, \ldots, c_s^l]^T$ and c^0 is a vector of 1's of length s. We will also say that the rth stage has stage order m (possibly higher than the overall stage order q) if the rth equation of SO_l is satisfied for $l = 1, \ldots, m$. The continuous order conditions, mentioned above, have two forms (see [19]): quadrature conditions,

$$b(\theta)^T c^{p-1} = \frac{c^p}{p}, \tag{11}$$

where p is a positive integer, and nonquadrature conditions, which can be described in terms of the stage order conditions (10). The nonquadrature conditions to be employed in this paper are of the form

$$\text{(a)}\ b(\theta)^T (SO_l) = 0, \quad \text{(b)}\ b(\theta)^T c(SO_l) = 0, \quad \text{(c)}\ b(\theta)^T X(SO_l) = 0, \tag{12}$$

$$b(\theta)^T c^2(SO_l) = 0, \quad b(\theta)^T cX(SO_l) = 0, \tag{13}$$

$$b(\theta)^T Xc(SO_l) = 0, \quad b(\theta)^T X^2(SO_l) = 0, \tag{14}$$

where l is a positive integer. Thus the basic idea is to choose the coefficients and weight polynomials of the CRK scheme (5), (8) to satisfy sufficient order conditions to obtain a scheme of a desired order, and this is the approach we shall employ in this paper. Some of the stage coefficients are determined by requiring the CRK scheme to include certain specific stages associated with the underlying collocation scheme.

The collocation polynomial associated with the collocation method can be expressed in a form alternative to that given in (3). On the ith subinterval, the collocation polynomial, $\hat{u}_i(t)$, can be viewed within the framework of the continuous Runge–Kutta schemes; it has the form

$$\hat{u}_i(t) = \hat{u}_i(t_{i-1} + \theta h) = y_{i-1} + h \sum_{r=1}^{k} \hat{b}_r(\theta) f(\widehat{t}_r, \widehat{y}_r), \tag{15}$$

where $t \in [t_{i-1}, t_i]$, $\hat{b}_r(\theta)$ is a polynomial in $\theta \in [0, 1]$; y_{i-1} is the discrete collocation solution approximation at t_{i-1},

$$\widehat{t}_r = t_{i-1} + \rho_r h, \quad \widehat{y}_r = y_{i-1} + h \sum_{j=1}^{s} a_{rj} f(\widehat{t}_j, \widehat{y}_j);$$

and the coefficients a_{rj} characterize the implicit Runge–Kutta scheme which is equivalent to the collocation scheme (see [24]). The collocation stages, $f(\widehat{t}_r, \widehat{y}_r)$, are the evaluations of the right-hand side of the BVODE that occur during the calculation of the collocation solution. The collocation stages have stage order k, i.e., the stage coefficients satisfy (10) with $q = k$. The weight polynomials, $\hat{b}_r(\theta)$, have derivatives that are the Lagrange interpolating polynomials for the abscissa set $\{\rho_r\}_{r=1}^{k}$.

A superconvergent interpolant is obtained from a CRK scheme which, on the ith subinterval, will use the superconvergent end point values y_{i-1} and y_i, the corresponding stages $f(t_{i-1}, y_{i-1})$ and $f(t_i, y_i)$, the collocation stages, and as many extra MIRK stages as necessary to achieve the desired order. Thus the first $k+2$ rows of the tableau of the CRK scheme, with the stages expressed in PIRK form (7), will be

$$\begin{array}{c|c|cccccccc} 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \rho_1 & 0 & 0 & 0 & a_{11} & \dots & a_{1k} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_k & 0 & 0 & 0 & a_{k1} & \dots & a_{kk} & 0 & \dots & 0 \end{array} . \tag{16}$$

Note that the collocation stages originally expressed in IRK form (6) can be written in PIRK form (7) by setting the extra free parameters $v_3, \dots, v_{k+2}$ to zero, as indicated in (16).

For the CRK scheme (5) assuming PIRK stages (7) there will be explicit dependence on y_{i-1} and on the derivative values at the end points of the subinterval. In subsequent sections of this paper we will exhibit CRK schemes, with weight polynomials derived from the order conditions, which interpolate the solution value y_{i-1} at the left end point and which have derivatives that interpolate $y'_{i-1} \equiv f(t_{i-1}, y_{i-1})$ and $y'_i \equiv f(t_i, y_i)$ at the left and right end points, respectively. Since there is no explicit dependence in (5) on y_i, the weight polynomials must satisfy special conditions for interpolation of y_i (see [23]). In the current setting these conditions are

$$b_1(1) = b_2(1) = 0, \quad b_r(1) = \hat{b}_{r-2}, r = 3, \dots, k+2, \quad b_r(1) = 0, r = k+3, \dots, s, \tag{17}$$

where $\hat{b}_r = \hat{b}_r(1)$ in (15). Under these conditions, the evaluation of the CRK scheme at the right-hand end point gives y_i in terms of the (discrete) implicit Runge–Kutta method equivalent to the collocation method. Then the CRK scheme interpolates y_i and the corresponding global interpolant will have C^1 continuity. This assumes that all of the collocation stages are included within the CRK scheme. However, in subsequent sections we shall demonstrate that although the conditions (17) are naturally satisfied by the higher order CRK schemes, derived only by applying the order conditions, some of the lower order schemes (see section 4 and subsection 5.1) do not satisfy them. This happens because the desired order can be obtained for these schemes without requiring the use of all of the collocation stages. Such schemes lead to an interpolant that is only "approximately" continuous to order $O(h^{2k})$. That is, in general there may be jump discontinuities in the interpolant, at the mesh points, of size $O(h^{2k})$. However, it is also possible to derive related CRK schemes which include all the collocation stages and which satisfy the conditions (17). For lower order CRK schemes, we will derive schemes using the minimum number of stages and having "approximate" continuity to order $(O(h^{2k})$ as well as schemes having an extra stage that allows them to have "exact" continuity. In either case the derivative will be continuous.

When a k-point Gaussian collocation scheme is used in the numerical solution of a BVODE, the discrete collocation solution will have a local error of $O(h^{2k+1})$ and a global error of $O(h^{2k})$ [4]. In order to examine the global error of the associated collocation polynomial, we begin by letting $y_i(t)$ be the true solution to the local problem on the ith subinterval (9), where the initial value y_{i-1} is the superconvergent solution value at t_{i-1}. Since the global error for y_{i-1} is $O(h^{2k})$, it follows that

$$|y_i(t) - y(t)| \quad \sim \quad O(h^{2k}),$$

i.e., the true local solution agrees with the true global solution to within a (global) error of $O(h^{2k})$. The collocation polynomial on the ith interval, $\hat{u}_i(t)$, agrees with the true local solution, $y_i(t)$, to within a local error of $O(h^{k+1})$, i.e.,

$$|y_i(t) - \hat{u}_i(t)| \quad \sim \quad O(h^{k+1}).$$

Thus for $k \geq 1$ we have

$$|\hat{u}_i(t) - y(t)| \quad \sim \quad O(h^{k+1}),$$

i.e., the global error of the collocation polynomial is $O(h^{k+1})$.

We now consider the CRK interpolants, similarly. Since the global error of the true local solution $y_i(t)$ is $O(h^{2k})$, it is sufficient for the CRK scheme to agree with $y_i(t)$ to within a local error of $O(h^{2k})$. Then the local error of the CRK scheme agrees with the global error of the true local solution and the global error of the CRK scheme will also be $O(h^{2k})$. On the other hand, it is sometimes useful, e.g., if the defect of the solution is important, to have an interpolant whose local error is dominated by the global error of the discrete solution. Thus in subsequent sections of this paper for each value of $k = 1, 2, 3, 4$ we will derive two CRK schemes, each having a global error of $O(h^{2k})$. The first scheme will have a local error of $O(h^{2k})$ while the second will have a local error of $O(h^{2k+1})$.

To summarize, we will derive CRK schemes by imposing specific order and stage order conditions on the coefficients and weights of the scheme. Some of the coefficients will be specified by requiring the CRK scheme to include end point and collocation stages. For some low order schemes, we will also impose continuity conditions (17) in order to ensure C^1 continuity for the interpolant.

4. Collocation with $k = 1$. A collocation scheme with collocation at one Gauss point per subinterval is equivalent to the 1-stage IRK scheme called the midpoint scheme. Its local error is $O(h^3)$; its global error is $O(h^2)$. The associated C^0 collocation polynomial written in CRK form (15) is

$$\hat{u}_i(t_{i-1} + \theta h) = y_{i-1} + \theta h f\left(t_{i-1} + \frac{h}{2}, \frac{y_{i-1} + y_i}{2}\right).$$

Its local and global error are both $O(h^2)$. Thus in this case the collocation polynomial is already "superconvergent."

A 2-stage CRK scheme (5) can be employed to provide an interpolant with global error $O(h^2)$ which has a continuous first derivative (unlike the collocation polynomial). We construct this scheme using the end point stages $f(t_{i-1}, y_{i-1})$ and $f(t_i, y_i)$. This determines the c, v, and x coefficients of the scheme. The weight polynomials are determined by imposing the continuous Runge–Kutta order conditions up to order 2, i.e., the two quadrature conditions (11) with $p = 1, 2$. This scheme has local error $O(h^3)$ and global error $O(h^2)$. Its tableau is

$$\begin{array}{c|c|cc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline & & \frac{\theta(2-\theta)}{2} & \frac{\theta^2}{2} \end{array}.$$

This scheme is the continuous version of the trapezoidal scheme. It does not use the collocation stage, and therefore the associated interpolant has only $O(h^2)$ continuity, although its derivative is continuous.

A C^1 continuous interpolant is obtained from a CRK scheme (5) whose three stages are the two end point stages and the collocation stage. The weight polynomials are determined from the order conditions (11) with $p = 1, 2$ and the continuity conditions (17). Its tableau is

$$\begin{array}{c|c|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \hline & & \theta(\theta-1)^2 & \theta^2(\theta-1) & \theta^2(3-2\theta) \end{array}.$$

This scheme has local error $O(h^3)$ and global error $O(h^2)$.

5. Collocation with $k = 2$. The implicit Runge–Kutta scheme corresponding to collocation at two Gauss points per subinterval is given in [7]. Its local error is $O(h^5)$ and its global error is $O(h^4)$. The stage order of each stage and thus of the whole scheme is 2. The associated C^0 collocation polynomial, written in CRK form (15), has the tableau

$$\begin{array}{c|c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & 0 & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & 0 & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & & (\frac{1+\sqrt{3}}{2})\theta - \frac{\sqrt{3}}{2}\theta^2 & (\frac{1-\sqrt{3}}{2})\theta + \frac{\sqrt{3}}{2}\theta^2 \end{array}.$$

The local and global errors for this scheme are both $O(h^3)$.

5.1. A superconvergent interpolant with local error $O(h^4)$. We now derive a CRK scheme (5) with local and global errors of $O(h^4)$ whose first two stages are the end point stages and whose third stage is one of the collocation stages. This determines the c, v, and x coefficients of the scheme. Since the stage order of the first two stages is 3 and the stage order of the third stage is 2, the stage order of the whole scheme is 2. The three weight polynomials are determined by requiring the scheme to satisfy the order conditions for order 3, which in this case are the three quadrature conditions (11) with $p = 1, 2, 3$. The tableau of the resultant scheme is

$$\begin{array}{c|c|cccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{3}}{6} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & 0 & 0 & 0 & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) \end{array}, \tag{18}$$

where

$$b_1(\theta) = \frac{(3+\sqrt{3})}{12}\theta(4\theta^2 - (9-\sqrt{3})\theta + 6 - 2\sqrt{3}),$$

$$b_2(\theta) = \frac{(3-\sqrt{3})}{12}\theta^2(4\theta - 3 + \sqrt{3}), \quad b_3(\theta) = \theta^2(3-2\theta),$$

and $b_4(\theta) \equiv 0$. (The scheme is written as a 4-stage scheme to clarify the representation of the collocation stages. Since the fourth weight is identically zero, the scheme can be implemented as a 3-stage scheme.) Since only one of the collocation stages is included, the associated interpolant will have a continuous derivative but itself will have only continuity to $O(h^4)$.

An interpolant with C^1 continuity is obtained by including the second collocation stage within (18), i.e., $b_4(\theta)$ must not be identically zero. Using the same tableau as in (18), the weight polynomials are determined by requiring them to satisfy the three quadrature conditions and the conditions (17). The resultant weight polynomials are

$$b_1(\theta) = \theta(\theta-1)^2, \quad b_2(\theta) = \theta^2(\theta-1),$$

and

$$b_3(\theta) = b_4(\theta) = \frac{1}{2}\theta^2(3-2\theta).$$

5.2. A superconvergent interpolant with local error $O(h^5)$. We conclude this section by deriving a C^1 interpolant based on a CRK scheme (5) with global error $O(h^4)$ and local error $O(h^5)$. The first two stages of the scheme will be the end point stages and the third and fourth stages will be the collocation stages. The stage order of the scheme is 2. The scheme must satisfy the order conditions for order 4, which in this case are the four quadrature conditions (11) with $p = 1, \ldots, 4$ and the nonquadrature condition (12)(a) with $l = 3$. Since these conditions are independent (as can be seen by examining the right-hand sides of the equations), the CRK scheme must employ five stages; thus we add a fifth MIRK stage, to which we apply the stage order 4 conditions. The embedding of the end point and collocation stages defines the first four components of c and v and the first four rows of x. The application of the stage order 4 conditions to the fifth stage defines x_{51}, x_{52}, x_{53}, and x_{54}, in terms of c_5 and v_5, and since the stage is monoimplicit, $x_{55} = 0$.

The weight polynomials are determined by solving the above order conditions and we obtain a 2-parameter family of CRK schemes, with c_5 and v_5 as the free parameters, with the restriction that $c_5 \neq 0, \frac{1}{2}, 1$. Since the expressions for the coefficients of the fifth stage and the weight polynomials are somewhat complicated, we do not give them here. An example of a specific CRK scheme from this family is obtained by choosing $c_5 = v_5 = \frac{2}{5}$. The tableau of the resultant scheme is

$$\begin{array}{c|c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} - \frac{\sqrt{3}}{6} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} & 0 \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & 0 & 0 & 0 & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} & 0 \\
\frac{2}{5} & \frac{2}{5} & \frac{36}{625} & -\frac{6}{625} & -\frac{3}{125} + \frac{54\sqrt{3}}{625} & -\frac{3}{125} - \frac{54\sqrt{3}}{625} & 0 \\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta)
\end{array},$$

where,

$$b_1(\theta) = \frac{1}{4}\theta(\theta-1)^2(5\theta+4), \quad b_2(\theta) = \frac{1}{3}\theta^2(\theta-1)(10\theta-7),$$

$$b_3(\theta) = -\frac{1}{2}\theta^2(3\theta-2)(5\theta-6), \quad b_4(\theta) = b_3(\theta), \quad b_5(\theta) = \frac{125}{12}\theta^2(\theta-1)^2.$$

6. Collocation with $k = 3$. The implicit Runge–Kutta scheme that is equivalent to collocation with three Gauss points is given in [7]. It has a local error of $O(h^7)$ and a global error of $O(h^6)$. Its stage order is 3. The corresponding C^0 collocation polynomial (15) has local and global errors of $O(h^4)$. It uses the three stages of the IRK scheme, and the weight polynomials have derivatives equal to the Lagrange interpolating polynomials for the abscissa of the IRK scheme. The collocation polynomial is presented in [15].

6.1. A superconvergent interpolant with local error $O(h^6)$. This C^1 superconvergent interpolant is based on a CRK scheme (5) with local and global errors of $O(h^6)$. Since the scheme will include the three collocation stages, its stage order will be 3. The scheme must satisfy the five quadrature conditions (11) with $p = 1, \ldots, 5$ and the nonquadrature condition (12)(a) with $l = 4$. An inspection of the right-hand sides of these conditions shows that they are independent and thus the CRK scheme will require six stages. In addition to the two end point stages and the three collocation stages, we will need one extra MIRK stage, which we will require to have stage order 5. The nonquadrature condition leads to the requirement that

$$b_5(\theta) = \frac{5}{4}b_4(\theta) - b_3(\theta).$$

This leaves $b_1(\theta)$, $b_2(\theta)$, $b_3(\theta)$, $b_4(\theta)$, and $b_6(\theta)$ to satisfy the remaining five quadrature conditions.

The two end point stages and the three collocation stages define the first five components of c and v and the first five rows of x. The imposition of stage order 5 on the sixth stage gives expressions for $x_{61}, \ldots, x_{65}$ in terms of c_6 and v_6 and since it is a monoimplicit stage, $x_{66} = 0$. The order conditions give expressions for $b_1(\theta)$, $b_2(\theta)$, $b_3(\theta)$, $b_4(\theta)$, and $b_6(\theta)$ in terms of the free parameters c_6 and v_6 with the restrictions that $c_6 \neq 0, 1$, or $\frac{1}{2} \pm \frac{\sqrt{5}}{10}$. Since the expressions for x_{61}, x_{62}, x_{63}, x_{64}, and x_{65} and for the weight polynomials are somewhat complicated we do not present them here. An example of a specific scheme from this family is obtained for the choice of $c_6 = v_6 = \frac{3}{5}$; we get the CRK scheme with tableau

$$\begin{array}{c|c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} - \frac{\sqrt{15}}{10} & 0 & 0 & 0 & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} & 0 \\
\frac{1}{2} & 0 & 0 & 0 & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} & 0 \\
\frac{1}{2} + \frac{\sqrt{15}}{10} & 0 & 0 & 0 & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} & 0 \\
\frac{3}{5} & \frac{3}{5} & -\frac{78}{3125} & \frac{72}{3125} & \frac{6\sqrt{15}}{125} - \frac{47}{1875} & \frac{488}{9375} & -\frac{6\sqrt{15}}{125} - \frac{47}{1875} & 0 \\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) & b_6(\theta)
\end{array},$$

where

$$b_1(\theta) = \frac{\theta\,(\theta-1)^2\left(5\,\theta^2-10\,\theta+3\right)}{3}, \quad b_2(\theta) = \frac{\theta^2\,(\theta-1)\left(30\,\theta^2-35\,\theta+9\right)}{4},$$

$$b_3(\theta) = \frac{(-5+\sqrt{15})}{36}(60\,\theta^3-15\,\sqrt{15}\theta^2-195\,\theta^2+220\,\theta+32\,\theta\sqrt{15}-18\,\sqrt{15}-90)\theta^2,$$

$$b_4(\theta) = -\frac{4\,\theta^2\left(-18+62\,\theta-75\,\theta^2+30\,\theta^3\right)}{9}, \quad b_6(\theta) = \frac{125\,\theta^2\,(2\,\theta-1)\,(\theta-1)^2}{12},$$

and

$$b_5(\theta) = -\frac{(5+\sqrt{15})}{36}(60\,\theta^3-195\,\theta^2+15\,\sqrt{15}\theta^2-32\,\theta\sqrt{15}+220\,\theta+18\,\sqrt{15}-90)\theta^2.$$

6.2. A superconvergent interpolant with local error $O(h^7)$. We conclude this section by considering a superconvergent C^1 interpolant based on a CRK scheme (5) with a local error of $O(h^7)$ and a global error of $O(h^6)$. Its stage order is 3 and it must satisfy 10 continuous order conditions: the 6 quadrature conditions (11) with $p = 1, \ldots, 6$ and the 4 nonquadrature conditions (12)(a) with $l = 4, 5$, (12)(b) with $l = 4$, and (12)(c) with $l = 4$.

In the analysis to follow, we will show that it is possible to satisfy the 10 order conditions using a CRK scheme with 8 stages. The resultant scheme will contain the end point stages and collocation stages embedded within it and will include 3 extra MIRK stages assumed to have at least stage order 5. We will follow a strategy similar to that of [19]. We focus on the 4 nonquadrature conditions. Since all stages except the third, fourth, and fifth have at least stage order 5, the first 3 nonquadrature conditions

reduce to

$$b_3(\theta)\left(\frac{1}{400}\right) + b_4(\theta)\left(-\frac{1}{320}\right) + b_5(\theta)\left(\frac{1}{400}\right) = 0,$$

$$b_3(\theta)\left(\frac{1}{800}-\frac{\sqrt{15}}{4000}\right) + b_4(\theta)\left(-\frac{1}{640}\right) + b_5(\theta)\left(\frac{1}{800}+\frac{\sqrt{15}}{4000}\right) = 0,$$

$$b_3(\theta)\left(\frac{1}{200}-\frac{3\sqrt{15}}{10000}\right) + b_4(\theta)\left(-\frac{1}{160}\right) + b_5(\theta)\left(\frac{1}{200}+\frac{3\sqrt{15}}{10000}\right) = 0.$$

These are all satisfied if

$$b_4(\theta) = \frac{8}{5}b_3(\theta) \quad \text{and} \quad b_5(\theta) = b_3(\theta).$$

We now return to the last of the nonquadrature conditions, namely,

$$b(\theta)^T X\left(Xc^3 + \frac{v}{4} - \frac{c^4}{4}\right) = 0.$$

Since all stages except the third, fourth, and fifth are chosen to have at least stage order 5, this condition reduces to

$$b_3(\theta)\left(\frac{\sqrt{15}}{8000}\right)+b_5(\theta)\left(-\frac{\sqrt{15}}{8000}\right)+b_6(\theta)\left(\frac{x_{63}}{400}-\frac{x_{64}}{320}+\frac{x_{65}}{400}\right)$$

$$+b_7(\theta)\left(\frac{x_{73}}{400}-\frac{x_{74}}{320}+\frac{x_{75}}{400}\right)+b_8(\theta)\left(\frac{x_{83}}{400}-\frac{x_{84}}{320}+\frac{x_{85}}{400}\right)=0.$$

Since $b_3(\theta)=b_5(\theta)$, this condition is satisfied if we choose

$$x_{r5}=\frac{5}{4}x_{r4}-x_{r3},\qquad r=6,7,8. \tag{19}$$

We can then determine the six weight polynomials $b_1(\theta)$, $b_2(\theta)$, $b_3(\theta)$, $b_6(\theta)$, $b_7(\theta)$, $b_8(\theta)$ by requiring them to satisfy the remaining six quadrature conditions.

The above strategy requires us to be able to impose the stage order 5 conditions as well as the conditions (19) on the last three stages. For the sixth stage, these requirements force $c_6=\frac{1}{2}$ and the parameter v_6 is left free. When we apply the stage order 5 conditions plus (19) to the seventh stage, we are left with c_7 and v_7 as free parameters. For the eighth stage, we apply stage order 6 (there are sufficient free parameters to allow higher stage order to be imposed) and (19), which leaves c_8 and v_8 free.

The weight polynomials are expressed in terms of the free parameters v_6, c_7, v_7, c_8, v_8 with the restrictions

$$c_7\neq 0,1,\frac{1}{2},c_8,1-c_8,\frac{1}{2}\pm\frac{\sqrt{15}}{10},\qquad c_8\neq 0,1,\frac{1}{2}. \tag{20}$$

Since the expressions for the coefficients of the last three stages and for the weight polynomials are somewhat complicated we do not give them here. An example of a scheme from this family is obtained for the choice of the free parameters, $v_6=\frac{1}{2}, c_7=\frac{1}{3}, v_7=\frac{1}{3}, c_8=\frac{2}{3}, v_8=\frac{2}{3}$; we get the scheme with tableau

$$\begin{array}{c|c|cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
c_3 & 0 & 0 & 0 & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} & 0 & 0 & 0\\
\frac{1}{2} & 0 & 0 & 0 & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} & 0 & 0 & 0\\
c_5 & 0 & 0 & 0 & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} & 0 & 0 & 0\\
\frac{1}{2} & \frac{1}{2} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 0 & 0 & 0\\
\frac{1}{3} & \frac{1}{3} & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & 0 & 0\\
\frac{3}{5} & \frac{3}{5} & x_{81} & x_{82} & x_{83} & x_{84} & x_{85} & x_{86} & x_{87} & 0\\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) & b_6(\theta) & b_7(\theta) & b_8(\theta)
\end{array},$$

where

$$c_3=\frac{1}{2}-\frac{\sqrt{15}}{10},\quad c_5=\frac{1}{2}+\frac{\sqrt{15}}{10},$$

$$x_{61} = -\frac{1}{32}, \quad x_{62} = \frac{1}{32}, \quad x_{63} = \frac{5\sqrt{15}}{96}, \quad x_{64} = 0, \quad x_{65} = -\frac{5\sqrt{15}}{96},$$

$$x_{71} = -\frac{2}{243}, \quad x_{72} = \frac{4}{243}, \quad x_{73} = \frac{25}{729} + \frac{10\sqrt{15}}{243},$$

$$x_{74} = \frac{40}{729}, \quad x_{75} = \frac{25}{729} - \frac{10\sqrt{15}}{243}, \quad x_{76} = -\frac{32}{243},$$

$$x_{81} = -\frac{1038}{15625}, \quad x_{82} = \frac{684}{15625}, \quad x_{83} = \frac{131}{20625} + \frac{2298\sqrt{15}}{34375},$$

$$x_{84} = \frac{1048}{103125}, \quad x_{85} = \frac{131}{20625} - \frac{2298\sqrt{15}}{34375}, \quad x_{86} = \frac{26208}{171875}, \quad x_{87} = -\frac{26244}{171875},$$

and the weight polynomials are

$$b_1(\theta) = \frac{\theta(\theta-1)^2(75\theta^3 - 63\theta^2 + 17\theta + 3)}{3},$$

$$b_2(\theta) = \frac{\theta^2(\theta-1)(125\theta^3 - 234\theta^2 + 149\theta + 36)}{4},$$

$$b_3(\theta) = -\frac{5}{18}\theta^2(300\theta^4 - 876\theta^3 + 945\theta^2 - 460\theta + 90),$$

$$b_4(\theta) = \frac{8}{5}b_3(\theta), \quad b_5(\theta) = b_3(\theta),$$

$$b_6(\theta) = -16\theta^2(\theta-1)^2(25\theta^2 - 26\theta + 9), \quad b_7(\theta) = \frac{81}{8}\theta^2(\theta-1)^2(25\theta^2 - 25\theta + 9),$$

$$b_8(\theta) = \frac{3125}{24}\theta^2(\theta-1)^2(3\theta^2 - 3\theta + 1).$$

7. Collocation with $k = 4$. In this section we consider collocation at four Gauss points per subinterval. The equivalent IRK scheme is given in [6]. This scheme has a local error of $O(h^9)$ and a global error of $O(h^8)$. The corresponding C^0 collocation polynomial (15) employs the four stages of this IRK scheme and has weight polynomials whose derivatives are the Lagrange interpolating polynomials for the abscissa of the implicit Runge–Kutta method. The complete collocation polynomial is presented in [15]. Its local and global errors are $O(h^5)$ and its stage order is 4.

7.1. A superconvergent interpolant with local error $O(h^8)$. We consider a C^1 interpolant based on a CRK scheme (5) having local and global errors of $O(h^8)$. We will assume the 2 end point stages and the 4 collocation stages are to be embedded and that the extra stages are required to have at least stage order 4 so that the stage order of the whole scheme will be 4. With this latter assumption, the desired scheme must satisfy 11 order conditions: the 7 quadrature conditions, (11) with $p = 1, \dots, 7$ and the 4 nonquadrature conditions (12)(a) with $l = 5, 6$, (12)(b) with $l = 5$, and (12)(c) with $l = 5$. In order to simplify the analysis, we make the further assumption that the new stages will have stage order 6. The complete derivation, which is a somewhat more complicated generalization of that given in the previous section, is given in [15]. Within that analysis, it is shown that it is possible to choose some of the coefficients to make 2 of the nonquadrature conditions dependent on the others, and thus the total number of stages required for this scheme is 9. The resultant family has 5 free parameters, c_8, c_9, v_8, v_9, and x_{98}. We conclude this case by presenting a specific CRK scheme obtained by setting $c_8 = \frac{1}{5}$ and $c_9 = \frac{4}{5}$, $v_8 = c_8$, $v_9 = c_9$, and $x_{98} = 0$. The tableau is

$$\begin{array}{c|c|ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_3 & 0 & 0 & 0 & x_{33} & x_{34} & x_{35} & x_{36} & 0 & 0 & 0 \\
c_4 & 0 & 0 & 0 & x_{43} & x_{44} & x_{45} & x_{46} & 0 & 0 & 0 \\
c_5 & 0 & 0 & 0 & x_{53} & x_{54} & x_{55} & x_{56} & 0 & 0 & 0 \\
c_6 & 0 & 0 & 0 & x_{63} & x_{64} & x_{65} & x_{66} & 0 & 0 & 0 \\
c_7 & v_7 & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & 0 & 0 & 0 \\
\frac{1}{5} & \frac{1}{5} & x_{81} & x_{82} & x_{83} & x_{84} & x_{85} & x_{86} & x_{87} & 0 & 0 \\
\frac{4}{5} & \frac{4}{5} & x_{91} & x_{92} & x_{93} & x_{94} & x_{95} & x_{96} & x_{97} & 0 & 0 \\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) & b_6(\theta) & b_7(\theta) & b_8(\theta) & b_9(\theta)
\end{array},$$

where c_3, c_4, c_5, c_6 are the abscissa and $x_{33}, \dots x_{36}, x_{43}, \dots x_{46}, x_{53}, \dots x_{56}, x_{63}, \dots x_{66}$, are the stage coefficients associated with the 4-stage, eighth order, implicit Runge–Kutta method given in [6] (see [15] for further details). Also,

$$c_7 = v_7 = \frac{1}{2} + \frac{\sqrt{7}}{14}, \quad x_{71} = \frac{3\sqrt{7}}{392} - \frac{3}{392}, \quad x_{72} = \frac{3\sqrt{7}}{392} + \frac{3}{392},$$

$$x_{73} = -\frac{5\sqrt{30}\sqrt{7}}{1176} - \frac{3\sqrt{30}\alpha_1}{10976} - \frac{3\sqrt{7}}{784} + \frac{27\alpha_1}{5488},$$

$$x_{74} = \frac{5\sqrt{30}\sqrt{7}}{1176} + \frac{3\sqrt{30}\alpha_2}{10976} - \frac{3\sqrt{7}}{784} + \frac{27\alpha_2}{5488},$$

$$x_{75} = \frac{5\sqrt{30}\sqrt{7}}{1176} - \frac{3\sqrt{30}\alpha_2}{10976} - \frac{3\sqrt{7}}{784} - \frac{27\alpha_2}{5488},$$

$$x_{76} = -\frac{5\sqrt{30}\sqrt{7}}{1176} + \frac{3\sqrt{30}\alpha_1}{10976} - \frac{3\sqrt{7}}{784} - \frac{27\alpha_1}{5488},$$

$$x_{81} = -\frac{992}{140625} + \frac{152\sqrt{7}}{140625}, \quad x_{82} = -\frac{628}{140625} + \frac{152\sqrt{7}}{140625},$$

$$x_{83} = \frac{452\,\alpha_1}{109375} + \frac{193\sqrt{30}}{18750} + \frac{152\sqrt{30}\sqrt{7}}{140625} - \frac{76\sqrt{7}}{9375} - \frac{226\sqrt{30}\,\alpha_1}{984375} + \frac{9}{3125},$$

$$x_{84} = \frac{452\,\alpha_2}{109375} - \frac{193\sqrt{30}}{18750} - \frac{152\sqrt{30}\sqrt{7}}{140625} - \frac{76\sqrt{7}}{9375} + \frac{226\sqrt{30}\,\alpha_2}{984375} + \frac{9}{3125},$$

$$x_{85} = -\frac{452\,\alpha_2}{109375} - \frac{193\sqrt{30}}{18750} - \frac{152\sqrt{30}\sqrt{7}}{140625} - \frac{76\sqrt{7}}{9375} - \frac{226\sqrt{30}\,\alpha_2}{984375} + \frac{9}{3125},$$

$$x_{86} = -\frac{452\,\alpha_1}{109375} + \frac{193\sqrt{30}}{18750} + \frac{152\sqrt{30}\sqrt{7}}{140625} - \frac{76\sqrt{7}}{9375} + \frac{226\sqrt{30}\,\alpha_1}{984375} + \frac{9}{3125},$$

$$x_{87} = \frac{4256\sqrt{7}}{140625}, \quad x_{91} = \frac{628}{140625} + \frac{152\sqrt{7}}{140625}, \quad x_{92} = \frac{992}{140625} + \frac{152\sqrt{7}}{140625},$$

$$x_{93} = -\frac{9}{3125} + \frac{452\,\alpha_1}{109375} - \frac{193\sqrt{30}}{18750} - \frac{76\sqrt{7}}{9375} - \frac{226\sqrt{30}\alpha_1}{984375} + \frac{152\sqrt{30}\sqrt{7}}{140625},$$

$$x_{94} = -\frac{9}{3125} + \frac{452\,\alpha_2}{109375} + \frac{193\sqrt{30}}{18750} - \frac{76\sqrt{7}}{9375} + \frac{226\sqrt{30}\alpha_2}{984375} - \frac{152\sqrt{30}\sqrt{7}}{140625},$$

$$x_{95} = -\frac{9}{3125} - \frac{452\,\alpha_2}{109375} + \frac{193\sqrt{30}}{18750} - \frac{76\sqrt{7}}{9375} - \frac{226\sqrt{30}\alpha_2}{984375} - \frac{152\sqrt{30}\sqrt{7}}{140625},$$

$$x_{96} = -\frac{9}{3125} - \frac{452\,\alpha_1}{109375} - \frac{193\sqrt{30}}{18750} - \frac{76\sqrt{7}}{9375} + \frac{226\sqrt{30}\alpha_1}{984375} + \frac{152\sqrt{30}\sqrt{7}}{140625},$$

$$x_{97} = \frac{4256\sqrt{7}}{140625}, \quad \alpha_1 = \sqrt{525 + 70\sqrt{30}}, \quad \alpha_2 = \sqrt{525 - 70\sqrt{30}}.$$

The expressions for the weight polynomials, even with the use of specific values for the free parameters, are very complicated. Thus we provide them in a simpler form by reducing the coefficients to 16 significant digits. This gives

$$b_1(\theta) = 118.2155391766658\theta^7 - 450.2127204516691\theta^6 + 677.9074502047085\theta^5$$

$$-506.6326577159307\theta^4 + 190.3161507410786\theta^3 - 30.59376195485317\theta^2 + 1.0\theta,$$

$$b_2(\theta) = -60.92387250999991\theta^7 + 249.6918871183337\theta^6 - 401.4282835380396\theta^5$$

$$+316.7368243825975\theta^4 - 122.3369840744121\theta^3 + 18.26042862151984\theta^2,$$

$$b_3(\theta) = -312.8902724924551\theta^7 + 1182.484831564545\theta^6 - 1760.013679899522\theta^5$$

$$+1290.0528958373388\theta^4 - 465.90057061724112\theta^3 + 66.440723029853349\theta^2,$$

$$b_4(\theta) = -305.59222210601167\theta^7 + 1134.2402328807773\theta^6 - 1652.2852242034485\theta^5$$

$$+1181.2977566097010\theta^4 - 415.39180675680098\theta^3 + 58.057336153267755\theta^2,$$

$$b_5(\theta) = 61.03778903256620\theta^7 - 278.29971712367288\theta^6 + 497.01008236447751\theta^5$$

$$-432.96119140493309\theta^4 + 182.24991389345502\theta^3 - 28.710804184452299\theta^2,$$

$$b_6(\theta) = 182.44470556590099\theta^7 - 725.92534732163222\theta^6 + 1143.7888217385517\theta^5$$

$$-890.88946104216607\theta^4 + 341.54246348060017\theta^3 - 50.787254998668808\theta^2,$$

$$b_7(\theta) = 214.91228070175543\theta^7 - 752.19298245614483\theta^6 + 1032.4385964912236\theta^5$$

$$-700.61403508771189\theta^4 + 238.12280701754432\theta^3 - 32.666666666666666\theta^2,$$

$$b_8(\theta) = 429.13801384671125\theta^7 - 1603.2561966116661\theta^6 + 2349.7734629640710\theta^5$$

$$-1685.4482584736616\theta^4 + 588.91084526808820\theta^3 - 79.117866993589911\theta^2,$$

$$b_9(\theta) = -326.34196121513336\theta^7 + 1243.4700124011131\theta^6 - 1887.1912261219963\theta^5$$

$$+1428.4581268946669\theta^4 - 537.51281895229927\theta^3 + 79.117866993589910\theta^2.$$

7.2. A superconvergent interpolant with local error $O(h^9)$. We conclude this section by considering a C^1 interpolant based on a CRK scheme (5) having local error $O(h^9)$ and global error $O(h^8)$. This scheme will include the 2 end point stages and the 4 collocation stages. We will choose the extra required stages to have stage order 7; this reduces the complexity of the order conditions that must be satisfied in order to derive the CRK scheme. With this assumption the stage order of the CRK scheme will be 4 and must satisfy 19 order conditions: the 8 quadrature conditions (11) with $p = 1, \dots, 8$ and the 11 nonquadrature conditions (12)(a) with $l = 5, 6, 7$, (12)(b) with $l = 6, 7$, (12)(c) with $l = 6, 7$, (13), (14). The complete derivation, which is again a more complicated generalization of that given in the previous section, is given in [15]. Within that analysis, it is shown that it is possible to choose some of the coefficients to make 5 of the nonquadrature conditions dependent on the others, and thus the total number of stages required for this scheme is 14. The resultant family has 20 free parameters, $c_r, r = 9, \dots, 14$, $v_r, r = 7, \dots, 14$, and $x_{rj}, j = 11, \dots, r-1, r = 12, \dots, 14$. An example of a specific scheme from this family is obtained by choosing the free parameters to be $v_r = c_r, r = 7, \dots, 14$,

$x_{rj} = 0, j = 11, \ldots, r-1, r = 12, \ldots, 14$, $c_9 = \frac{1}{8}$, $c_{10} = \frac{1}{4}$, $c_{11} = \frac{3}{4}$, $c_{12} = \frac{7}{8}$, $c_{13} = \frac{2}{5}$, $c_{14} = \frac{3}{5}$. The tableau of this CRK scheme is

$$
\begin{array}{c|c|ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\
c_3 & 0 & 0 & 0 & x_{33} & x_{34} & x_{35} & x_{36} & 0 & \ldots & 0 \\
c_4 & 0 & 0 & 0 & x_{43} & x_{44} & x_{45} & x_{46} & 0 & \ldots & 0 \\
c_5 & 0 & 0 & 0 & x_{53} & x_{54} & x_{55} & x_{56} & 0 & \ldots & 0 \\
c_6 & 0 & 0 & 0 & x_{63} & x_{64} & x_{65} & x_{66} & 0 & \ldots & 0 \\
\frac{1}{2} & \frac{1}{2} & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & 0 & \ldots & 0 \\
c_8 & v_8 & x_{81} & x_{82} & x_{83} & x_{84} & x_{85} & x_{86} & x_{87} & \ldots & 0 \\
\frac{1}{8} & \frac{1}{8} & x_{91} & x_{92} & x_{93} & x_{94} & x_{95} & x_{96} & x_{97} & \ldots & 0 \\
\frac{1}{4} & \frac{1}{4} & x_{10,1} & x_{10,2} & x_{10,3} & x_{10,4} & x_{10,5} & x_{10,6} & x_{10,7} & \ldots & 0 \\
\frac{3}{4} & \frac{3}{4} & x_{11,1} & x_{11,2} & x_{11,3} & x_{11,4} & x_{11,5} & x_{11,6} & x_{11,7} & \ldots & 0 \\
\frac{7}{8} & \frac{7}{8} & x_{12,1} & x_{12,2} & x_{12,3} & x_{12,4} & x_{12,5} & x_{12,6} & x_{12,7} & \ldots & 0 \\
\frac{2}{5} & \frac{2}{5} & x_{13,1} & x_{13,2} & x_{13,3} & x_{13,4} & x_{13,5} & x_{13,6} & x_{13,7} & \ldots & 0 \\
\frac{3}{5} & \frac{3}{5} & x_{14,1} & x_{14,2} & x_{14,3} & x_{14,4} & x_{14,5} & x_{14,6} & x_{14,7} & \ldots & 0 \\
\hline
 & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) & b_6(\theta) & b_7(\theta) & \ldots & b_{14}(\theta)
\end{array},
$$

where the coefficients for the third through sixth stages, $c_3, \ldots c_6$ and $x_{33}, \ldots x_{36}$, $x_{43}, \ldots x_{46}$, $x_{53}, \ldots x_{56}$, $x_{63}, \ldots x_{66}$, are the stage coefficients associated with the 4-stage, eighth order, implicit Runge–Kutta method given in [6]. The expressions for the remaining coefficients and the weight polynomials, even with the use of specific values for the free parameters, are very complicated. Thus we provide them in a simpler form by reducing the coefficients to 16 significant digits. This gives

$$x_{71} = -0.005208333333333333, \quad x_{72} = 0.005208333333333333, \quad x_{73} = 0.09768756687169055,$$

$$x_{74} = 0.1355546821755248, \quad x_{75} = -0.1355546821755248, \quad x_{76} = -0.09768756687169055$$

for the seventh stage;

$$x_{81} = -0.008066287689150030, \quad x_{82} = 0.007239834759829562, \quad x_{83} = 0.1371075768778618,$$

$$x_{84} = 0.09571675661727026, \quad x_{85} = -0.05699647651059596, \quad x_{86} = -0.06921542910219575,$$

$$x_{87} = -0.1057859749530199$$

for the eighth stage;

$$x_{91} = 0.01924728724952684, \quad x_{92} = 0.0009231770064713210, \quad x_{93} = 0.08269651783093607,$$

$$x_{94} = 0.04408472965941152, \quad x_{95} = -0.03465768595451309, \quad x_{96} = -0.02368831572171980,$$

$$x_{97} = -0.01629234729465400, \quad x_{98} = -0.07231336277545893$$

for the ninth stage;

$$x_{10,1} = 0.04018535463530762, \quad x_{10,2} = -0.0007184435193296687, \quad x_{10,3} = -0.05149956819838338,$$

$$x_{10,4} = -0.09119800053831042, \quad x_{10,5} = -0.08421585380834320, \quad x_{10,6} = -0.04206634835217353,$$

$$x_{10,7} = 0.001968952482382824, \quad x_{10,8} = 0.03646061187208010, \quad x_{10,9} = 0.1910832954267680$$

for the tenth stage;

$$x_{11,1} = -0.06747400064574176, \quad x_{11,2} = 0.01129454483158386, \quad x_{11,3} = 0.3416237837455131,$$

$$x_{11,4} = 0.3601250455762746, \quad x_{11,5} = -0.005638214280780879, \quad x_{11,6} = -0.1525401711723007,$$

$$x_{11,7} = 0.0, \quad x_{11,8} = -0.3468850433707371, \quad x_{11,9} = -0.3290534392809501,$$

$$x_{11,10} = 0.1885474945972381$$

for the eleventh stage;

$$x_{12,1} = -0.04536130701043499, \quad x_{12,2} = -0.01881103880965994, \quad x_{12,3} = 0.2633344008411732,$$

$$x_{12,4} = 0.2972755846582715, \quad x_{12,5} = 0.04101039168870139, \quad x_{12,6} = -0.08289233968810006,$$

$$x_{12,7} = 0.0, \quad x_{12,8} = -0.4889108696816266, \quad x_{12,9} = -0.3269404305894942,$$

$$x_{12,10} = 0.3612956085915711$$

for the twelfth stage;

$$x_{13,1} = 0.03608279120121541, \quad x_{13,2} = 0.000131678572969903, \quad x_{13,3} = -0.06639084080129527,$$

$$x_{13,4} = -0.1264523666751095, \quad x_{13,5} = -0.1290426551933866, \quad x_{13,6} = -0.06989044658201478,$$

$$x_{13,7} = 0.0, \quad x_{13,8} = 0.2096860531012513, \quad x_{13,9} = 0.1918513160723896,$$

$$x_{13,10} = -0.04597552969601728$$

for the thirteenth stage; and

$$x_{14,1} = -0.02178576193711737, \quad x_{14,2} = 0.01258806317607745, \quad x_{14,3} = 0.1140050201965098,$$

$$x_{14,4} = 0.07482936616593614, \quad x_{14,5} = -0.1064001928984542, \quad x_{14,6} = -0.1308449291677193,$$

$$x_{14,7} = 0.0, \quad x_{14,8} = 0.2764275204217753, \quad x_{14,9} = 0.02735915640638052,$$

$$x_{14,10} = -0.2461782423633799$$

for the fourteenth stage.

The weight polynomials are

$$b_1(\theta) = -56.44363811998998809\theta^8 + 253.8122387156809954\theta^7 - 472.8303146326085563\theta^6$$
$$+ 470.6933663638888841\theta^5 - 268.7588196996464491\theta^4$$
$$+ 87.35943397959582204\theta^3 - 14.83226660692058816\theta^2 + 1.0\theta,$$

$$b_2(\theta) = -5.696107732450688821\theta^8 + 50.83646998900248811\theta^7 - 132.0579204347868811\theta^6$$
$$+ 158.7585924649729902\theta^5 - 99.72388296679521135\theta^4$$
$$+ 32.39796672751068884\theta^3 - 4.515118047453358881\theta^2,$$

$$b_3(\theta) = 139.5694629392165599\theta^8 - 619.0329051265372271\theta^7 + 1130.299160884714430\theta^6$$
$$- 1090.735595491199596\theta^5 + 590.8363598357767708\theta^4$$
$$- 173.2626835821734419\theta^3 + 22.50012796356777711\theta^2,$$

$$b_4(\theta) = 261.6595694868406656\theta^8 - 1160.539562469618816\theta^7 + 2119.042271855944448\theta^6$$
$$- 2044.869990398276639\theta^5 + 1107.677741936499558\theta^4$$
$$- 324.8263498297635562\theta^3 + 42.18239199580467769\theta^2,$$

$$b_5(\theta) = 261.6595694868406656\theta^8 - 1160.539562469618816\theta^7 + 2119.042271855944448\theta^6$$
$$- 2044.869990398276639\theta^5 + 1107.677741936499558\theta^4$$
$$- 324.8263498297635562\theta^3 + 42.18239199580467769\theta^2,$$

$$b_6(\theta) = 139.5694629392165599\theta^8 - 619.0329051265372271\theta^7 + 1130.299160884714430\theta^6$$
$$- 1090.735595491199596\theta^5 + 590.8363598357767708\theta^4$$

$$- 173.2626835821734193\theta^3 + 22.50012796356777711\theta^2,$$

$$b_7(\theta) \equiv 0,$$

$$b_8(\theta) = -8.311419091396759640^8 + 35.83221757660262252\theta^7 - 62.94761883600197762\theta^6$$
$$+ 57.52243486393713397\theta^5 - 28.74509465841199015\theta^4$$
$$+ 7.420786734403720070^3 - 0.7713065891248477742\theta^2,$$

$$b_9(\theta) = 7.273474289193182170^8 - 31.35742649988263380\theta^7 + 55.08660821152866631\theta^6$$
$$- 50.33893086533372225\theta^5 + 25.15535609976147220^4$$
$$- 6.494065685382331570^3 + 0.6749844501153747870^2,$$

$$b_{10}(\theta) = -386.3626011670960410^8 + 1665.687728683040490^7 - 2926.167659615087580^6$$
$$+ 2673.973880405145920^5 - 1336.237460882881390^4$$
$$+ 344.9608825980491040^3 - 35.854770021170508 6\theta^2,$$

$$b_{11}(\theta) = 259.0733994058220630^8 - 914.5865444844163580^7 + 1241.882119998521320^6$$
$$- 802.5723636692900092\theta^5 + 240.3178383388930073\theta^4$$
$$- 21.95473967685173348\theta^3 - 2.159709912678271163\theta^2,$$

$$b_{12}(\theta) = -290.8101667566514860^8 + 1160.804774052110730^7 - 1902.767040748453150^6$$
$$+ 1651.047992616718893\theta^5 - 809.1141173845920550^4$$
$$+ 216.9895501921951770^3 - 26.15099197132814570^2,$$

$$b_{13}(\theta) = 222.0200497263612224\theta^8 - 862.2786617436748370^7 + 1301.057007450722860^6$$
$$- 933.7097504601227755\theta^5 + 296.2267532661724540^4$$
$$- 16.27927459466120059\theta^3 - 7.036123644797732110^2,$$

$$b_{14}(\theta) = -543.2010554059060230^8 + 2200.394138903847420^7 - 3599.938046875152320^6$$
$$+ 3045.835950060627540^5 - 1416.148775657044520^4$$
$$+ 351.7775265490147240^3 - 38.719737575386824 7\theta^2.$$

8. Numerical results. In this section we present numerical results for implementations of superconvergent interpolants based on several of the CRK schemes derived earlier. We consider collocation with $k = 2, 3$, and 4, and, for each value of k, we consider the corresponding superconvergent interpolant based on the CRK scheme having local error $O(h^{2k})$. (For the $k = 2$ case we consider the interpolant that has continuity to $O(h^4)$.) The double precision version of COLNEW was used to compute all collocation solutions. These tests were run on a DEC alpha system under the DEC UNIX operating system using the DEC UNIX Fortran 77 compiler.

8.1. Rates of convergence. In this subsection we present results which numerically investigate the orders of convergence for the discrete collocation solution, the collocation polynomial, and the superconvergent interpolant. We consider uniform meshes having N, the number of subintervals, equal to 4, 8, 16, 32, 64, 128, and 256.

The test problem is Swirling Flow III [2] (with $\epsilon = 0.075$):

$$\epsilon f''''(t) + f(t)f'''(t) + g(t)g'(t) = 0, \quad \epsilon g''(t) + f(t)g'(t) - f'(t)g(t) = 0,$$

$$f(0) = f(1) = f'(0) = f'(1) = 0, \quad g(0) = 1, \quad g(1) = -1.$$

(Since no closed form solution is available the COLNEW code was run in quadruple precision with a stringent tolerance to obtain a high accuracy reference solution.) For each solution component, the initial solution approximation is a straight line through the boundary conditions for components with boundary conditions and zero otherwise. Other test problems have also been considered; see [15] for further examples.

To obtain discrete solutions on specific uniform meshes, COLNEW was run with the communication parameters IPAR(8)=2 and IPAR(10)=2. When the corresponding Newton iterations converged this meant that COLNEW would (i) compute a solution on the given initial mesh, then (ii) generate a new mesh by doubling the number of mesh points, then (iii) compute a second solution on the mesh from step (ii), (iv) generate an error estimate, and then (v) terminate. This helped us to control the meshes used by COLNEW so that comparisons of solutions obtained from meshes of N and $2N$ subintervals, for experimental analysis of convergence rates, would be possible.

The discrete solution values at the mesh points were obtained by subsequent calls to the APPSLN routine, which is part of the COLNEW package. The global error for the corresponding collocation polynomial was based on evaluating the approximate solution, through calls to the APPSLN routine, at a large number of nonmesh sample points on each subinterval. Once the COLNEW code has computed a numerical solution, it is straightforward to employ the CRK schemes derived in the previous sections of this paper to construct a high order superconvergent interpolant. The algorithm will be discussed in detail in the next subsection. Subsequent to the execution of this algorithm, the superconvergent interpolant can be evaluated through calls to the INTERPEVAL routine, from the MIRKDC package. The global error estimate for the superconvergent interpolant was based on calls to the INTERPEVAL routine at a large number of sample points on each subinterval.

In Tables 2, 3, and 4, we give the ratio, for each mesh, of the maximum value of the global error of the approximate solution computed on the previous mesh to that for the current mesh, for the discrete collocation solution, the collocation polynomial, and the superconvergent interpolant. For each ratio, we also give in brackets the maximum value of the global error. For a given k, the rates of convergence predicted

TABLE 2
Error ratios (maximum errors): $k = 2$.

N	Discrete solution	Collocation polynomial	Superconvergent interpolant
4	14.4(7.2x10^{-3})	6.7(3.5x10^{-2})	5.9(5.5x10^{-2})
8	19.7(3.7x10^{-4})	5.5(6.3x10^{-3})	9.9(5.6x10^{-3})
16	16.8(2.2x10^{-5})	5.9(1.1x10^{-3})	12.7(4.4x10^{-4})
32	16.2(1.3x10^{-6})	6.9(1.5x10^{-4})	14.0(3.1x10^{-5})
64	16.1(8.3x10^{-8})	7.5(2.1x10^{-5})	14.4(2.2x10^{-6})
128	16.0(5.2x10^{-9})	7.7(2.7x10^{-6})	15.2(1.4x10^{-7})
256	16.0(3.3x10^{-10})	7.9(3.4x10^{-7})	15.6(9.1x10^{-9})
Theoretical	16	8	16

TABLE 3
Error ratios (maximum errors): $k = 3$.

N	Discrete solution	Collocation polynomial	Superconvergent interpolant
4	24.5(3.8x10^{-4})	4.7(7.7x10^{-3})	11.7(1.3x10^{-3})
8	41.7(9.1x10^{-6})	7.9(9.8x10^{-4})	31.3(4.2x10^{-5})
16	54.9(1.7x10^{-7})	11.2(8.7x10^{-5})	40.7(1.0x10^{-6})
32	61.3(2.7x10^{-9})	13.4(6.5x10^{-6})	50.5(2.1x10^{-8})
64	63.3(4.3x10^{-11})	14.7(4.4x10^{-7})	56.7(3.6x10^{-10})
128	63.1(6.8x10^{-13})	15.3(2.9x10^{-8})	60.1(6.1x10^{-12})
256	34.7(2.0x10^{-14})	15.7(1.8x10^{-9})	41.6(1.5x10^{-13})
Theoretical	64	16	64

by theory for these three solution approximations are, respectively, $O(h^{2k})$, $O(h^{k+1})$, and $O(h^{2k})$.

Note that as the accuracy of the approximate solutions begins to approach that of the reference solution (which is close to the double precision roundoff level), the estimate of the global error becomes unreliable, and thus the estimates of the convergence rates become irrelevant.

From Tables 2–4, we can observe that the expected rates of convergence are being approached by the numerical approximations and that the maximum error for the superconvergent interpolant is usually about an order of magnitude larger than that of the discrete collocation solution. A similiar situation has been examined in Table 3 ($\epsilon = 0.0001$ case) of [21, p. 723], where it was observed that the selection of the free parameters of a scheme can have a significant effect on the accuracy of the superconvergent interpolant. Table 3 of [21] shows superconvergent interpolant errors, for reasonable choices of abscissa, that are, in some instances, one to two orders of magnitude larger than the discrete solution error. The authors identify the optimal selection of the secondary collocation abscissa as an important subject for future investigation. For the schemes derived in this paper, the free parameters are chosen somewhat arbitrarily, simply to avoid singularities arising in the expressions for the coefficients and weight polynomials. We expect that somewhat more accurate superconvergent interpolants could be obtained by a detailed analysis of the principal error coefficients of the CRK schemes and an optimal selection of free parameters. See [7] for the general approach and [17] for an example of its application in the derivation of optimal MIRK and continuous MIRK schemes.

8.2. Execution times. In this subsection we consider execution time results for the calculation and evaluation of the collocation solution and the setup and evaluation of the associated superconvergent interpolant. The initial mesh is a uniform mesh of

TABLE 4
Error ratios (maximum errors): $k = 4$.

N	Discrete solution	Collocation polynomial	Superconvergent interpolant
4	170.4(1.1x10^{-5})	10.9(1.5x10^{-3})	73.9(6.5x10^{-5})
8	138.1(8.1x10^{-8})	18.4(8.1x10^{-5})	71.6(9.1x10^{-7})
16	203.9(4.0x10^{-10})	24.8(3.3x10^{-6})	136.6(6.7x10^{-9})
32	241.8(1.6x10^{-12})	28.6(1.1x10^{-7})	188.7(3.5x10^{-11})
64	205.0(8.0x10^{-15})	30.4(3.8x10^{-9})	64.1(5.5x10^{-13})
128	0.9(9.3x10^{-15})	31.7(1.2x10^{-10})	2.6(2.1x10^{-13})
256	1.0(9.8x10^{-15})	32.2(3.7x10^{-12})	2.6(9.0x10^{-14})
Theoretical	256	32	256

five subintervals and a tolerance of 10^{-6} is applied to each solution component. We use the test problem from the previous subsection. The COLNEW code ran to completion, using a sequence of meshes and Newton iterations, until a converged solution within the tolerance was obtained.

The construction of the superconvergent interpolant is based on the following algorithm which assumes that COLNEW has already been used to compute an approximate solution.

ALGORITHM FOR THE CONSTRUCTION OF A SUPERCONVERGENT INTERPOLANT.

1. The APPSLN routine, from the COLNEW package, is used to evaluate the solution at the mesh points and at the k collocation points within each subinterval.
2. The user-defined FSUB function is then called to compute the end point and collocation stages from the solution values computed in the previous step.
3. The INTERPSETUP routine from the MIRKDC package is called, once for each subinterval, to set up the extra MIRK stages, as specified by the CRK scheme corresponding to the value of k, identified in the previous sections of this paper.

In step 1, by calling the APPSLN routine we obtain the mesh point solution values and the arguments for the collocation stages, which are then computed in step 2. A somewhat more efficient implementation would involve recovery of the mesh point values and the collocation stages by directly accessing them from the FSPACE array retuned by the COLNEW code. This would avoid $k + 2$ calls to the APPSLN and k calls to the FSUB routine for each subinterval. However, the approach implemented here makes use of the interface routine provided by COLNEW and thus is perhaps somewhat simpler and, as we will show shortly, introduces negligible extra cost. Upon completion of the above algorithm, the superconvergent interpolant is then available for evaluation. It can be evaluated using a call to the INTERPEVAL routine from the MIRKDC package.

The important execution costs here are (i) the original call to COLNEW, which yields the discrete solution on the final mesh of N subintervals, and (ii) the construction of the superconvergent interpolant, using the above algorithm, in which the costs are attributable largely to calls to the APPSLN routine (step 1) and the INTERPSETUP routine (step 3). (For each call to the APPSLN routine there is a subsequent call to the user-defined FSUB function (step 2) but the latter cost is usually negligible.)

Another concern is the cost associated with the subsequent evaluation of the superconvergent interpolant. Since the superconvergent interpolant typically employs about twice as many stages as the collocation polynomial, we expect its evaluation cost

TABLE 5
Size of final mesh and total execution times.

k	N	COLNEW	APPSLN	INTERPSETUP	TOTAL SETUP
2	160	0.35	8.3x10^{-3}	-	8.3x10^{-3}
3	40	0.14	2.1x10^{-3}	7.0x10^{-4}	2.8x10^{-3}
4	20	0.08	1.4x10^{-3}	1.1x10^{-3}	2.5x10^{-3}

TABLE 6
Execution time for a single call.

k	APPSLN	INTERPEVAL
2	1.7x10^{-5}	1.7x10^{-5}
3	1.3x10^{-5}	3.0x10^{-5}
4	1.4x10^{-5}	3.7x10^{-5}

to be roughly twice that of the collocation polynomial; i.e., a call to the INTERPEVAL routine will likely cost about twice as much as a call to the APPSLN routine.

In Table 5, for $k = 2, 3$, and 4 we list execution times (in seconds) for the initial calculation of the collocation solution in the column labeled COLNEW and for each of the two main steps of the algorithm for the construction of the superconvergent interpolant. The times for step 1, which consist of $N + 1 + kN$ calls to the APPSLN routine, are given in the column labeled APPSLN, and those for step 3, which consist of N calls to the INTERPSETUP routine, are in the column labeled INTERPSETUP. The column labeled TOTAL SETUP gives the sum of the previous two columns and represents the total setup time for the construction of the superconvergent interpolant. We also give the number of subintervals in the final mesh required by COLNEW to compute its solution in the column labeled N. Note that for $k = 2$, no extra stages are needed for the superconvergent interpolant and thus no calls to the INTERPSETUP routine are required.

From Table 5, one can observe that the significant execution costs associated with setting up the superconvergent interpolant are negligible with respect to those associated with the original computation of the discrete solution. For $k = 2, 3$, and 4, the total setup costs represent approximately 2.4%, 2.0%, and 3.1%, respectively, of the cost associated with the calculation of the original collocation solution by COLNEW.

In Table 6 we give the execution time for a single call to APPSLN and a single call to INTERPEVAL; we observe that the cost associated with an evaluation of the superconvergent interpolant is within a factor of at most 2.6 times that of the collocation polynomial.

Since the execution costs for COLNEW are $O(Nn^3k^3)$ while the costs for computing the superconvergent interpolant are $O(Nnk^2)$, for larger values of k, it is clear that the cost of constructing a superconvergent interpolant will continue to be a negligible fraction of the cost of computing the discrete solution.

In Table 7, we provide maximum errors for the discrete collocation solution, the collocation polynomial, and the superconvergent interpolant for the cases considered in Table 5. These results show that COLNEW yields a continuous solution that approximately meets the tolerance, 10^{-6}, while the discrete solution and the superconvergent interpolant are substantially more accurate.

In Table 8, we provide maximum errors for the discrete collocation solution, the collocation polynomial, and the superconvergent interpolant for the $k = 2, 3$, and 4 when the collocation solution is computed on a nonuniform mesh. The mesh was

Table 7
Size of final mesh and maximum error.

k	N	Discrete solution	Collocation polynomial	Superconvergent interpolant
2	160	$2.1 \text{x} 10^{-9}$	$1.4 \text{x} 10^{-6}$	$5.9 \text{x} 10^{-8}$
3	40	$7.1 \text{x} 10^{-10}$	$2.8 \text{x} 10^{-6}$	$5.7 \text{x} 10^{-9}$
4	20	$6.8 \text{x} 10^{-11}$	$1.1 \text{x} 10^{-6}$	$1.3 \text{x} 10^{-9}$

Table 8
Size of final nonuniform mesh and maximum error.

k	N	Discrete solution	Collocation polynomial	Superconvergent interpolant
2	640	$3.1 \text{x} 10^{-10}$	$3.3 \text{x} 10^{-7}$	$8.8 \text{x} 10^{-9}$
3	160	$4.0 \text{x} 10^{-11}$	$4.2 \text{x} 10^{-7}$	$3.4 \text{x} 10^{-10}$
4	40	$3.7 \text{x} 10^{-10}$	$3.1 \text{x} 10^{-6}$	$6.2 \text{x} 10^{-9}$

constructed to cluster a large percentage of mesh points near the center of the problem interval. The maximum local mesh ratio is 100; the maximum global mesh ratio is 4952. Results are similar to those for Table 7, where uniform meshes were considered, indicating as expected from the theory that the accuracy of the superconvergent interpolants is not dependent on mesh ratios.

In Table 9, we provide maximum errors for the discrete collocation solution, the collocation polynomial, and the superconvergent interpolant for more difficult versions of the test problem, with smaller ϵ values, whose solutions exhibit more severe boundary layers.

These results generally agree with those of previous tables. The superconvergent interpolant is substantially more accurate than the collocation polynomial and is usually within about an order of magnitude of the discrete solution, even for problems with severe boundary layers. We were unable to present results for $k = 2$ for any of these values of ϵ or for $k = 4$ when $\epsilon = 0.0005$ because COLNEW was not able to obtain a solution (due to nonconvergent Newton iterations). We note that for the problem with the very severe boundary layer ($\epsilon = 0.0005$) the discrete solution is no longer substantially more accurate than the collocation polynomial. This of course affects the possible relative improvement of the superconvergent interpolant compared to the collocation polynomial. We also note that for problems in which a very high tolerance is applied, the resultant collocation polynomial solution will be very accurate and roundoff will limit the superconvergent discrete solution from being substantially more accurate. Again, this will mean that the superconvergent interpolant cannot be substantially more accurate than the collocation polynomial.

9. Summary, conclusions, and future work. In this paper, the question of developing an efficient method for augmenting the high order discrete solution yielded by a collocation method is explored. It is observed that it is possible to make use of the solution values at the mesh points and derivative values at the collocation points (information that is naturally produced during the collocation calculation) in the construction of a C^1 continuous interpolant to the high order mesh point solution. The approach is based on the use of CRK schemes, which incorporate the available information from the collocation computation and augment it with additional derivative information obtained using efficiently computable MIRK stages. C^1 superconvergent interpolants having global errors of $O(h^{2k})$ based on two CRK schemes with local errors of $O(h^{2k})$ and $O(h^{2k+1})$, respectively, are derived for k, the number of col-

TABLE 9
Size of final nonuniform mesh and maximum error for small ϵ.

ϵ	k	N	Discrete solution	Collocation polynomial	Superconvergent interpolant
0.002	3	180	$1.2x10^{-9}$	$1.5x10^{-5}$	$1.6x10^{-8}$
	4	80	$2.4x10^{-9}$	$5.1x10^{-5}$	$9.9x10^{-8}$
0.001	3	320	$9.3x10^{-10}$	$1.1x10^{-5}$	$5.9x10^{-9}$
	4	80	$1.8x10^{-8}$	$3.2x10^{-4}$	$9.4x10^{-7}$
0.0005	3	440	$2.2x10^{-8}$	$2.5x10^{-6}$	$2.2x10^{-8}$

location points, taking on the values 1, 2, 3, and 4. An alternative approach for deriving CRK schemes, based on the use of a boot-strapping algorithm, is also discussed.

Although the derivation process is more tedious, the direct derivation of CRK schemes, in which the coefficients which define the scheme are determined by requiring them to satisfy continuous order conditions, leads to schemes which use fewer stages and thus are more efficient than those obtained from the boot-strapping algorithm, at least for the values of k considered in this paper. The implementation or construction costs for these interpolants, when compared with the cost of computing the original discrete solution using a collocation method, are demonstrated through some numerical tests to be negligible. The cost of evaluating these interpolants is low; it is about twice that of evaluating the lower order collocation polynomial produced by the collocation process. Since the algorithm can be implemented through calls to existing routines from the COLNEW and MIRKDC codes, the software development effort associated with preparing the software that performs the construction and evaluation of the superconvergent interpolants is also quite reasonable. The numerical testing indicates that we can expect the accuracy of the superconvergent interpolant to be within about an order of magnitude of that of the discrete solution. Furthermore, it indicates that the accuracy of the superconvergent interpolant is not particularly sensitive to the use of nonuniform meshes or the presence of a numerical solution which exhibits severe boundary layers.

There is a variety of directions for future work. One is to continue the development of these interpolants for higher values of k, at least as far as $k = 7$ (the maximum value supported by the COLNEW package). Another is to investigate the development of superconvergent interpolants based on generalizations of CRK schemes or boot strapping for higher order differential equations, since the collocation method is able to handle such equations quite naturally. A third area of investigation involves modifying the COLNEW code so that it computes a superconvergent interpolant which it then uses for global error estimation and mesh selection.

Acknowledgments. The authors would like to thank Graeme Day for his assistance with the numerical testing aspects of this work. The authors are also grateful to the referees for their many helpful suggestions.

REFERENCES

[1] U.M. Ascher, J. Christiansen, and R.D. Russell, *Collocation software for boundary value ODEs*, ACM Trans. Math. Software, 7 (1981), pp. 209–222.

[2] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics 13, SIAM, Philadelphia, PA, 1995.

[3] G. Bader and U.M. Ascher, *A new basis implementation for a mixed order boundary value ODE solver*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 483–500.

[4] C. de Boor and B. Swartz, *Collocation at Gaussian points*, SIAM J. Numer. Anal., 10 (1973), pp. 582–606.

[5] K. Burrage, F.H. Chipman, and P.H. Muir, *Order results for mono-implicit Runge–Kutta methods*, SIAM J. Numer. Anal., 31 (1994), pp. 876–891.

[6] J.C. Butcher, *Implicit Runge–Kutta processes*, Math. Comp., 18 (1964), pp. 50–64.

[7] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, Wiley, Chichester, UK, 1987.

[8] J.R. Cash, *On the numerical integration of nonlinear two-point boundary value problems using iterated deferred corrections, Part* 1*: A survey and comparison of some one-step formulae*, Comput. Math. Appl., 12a (1986), pp. 1029–1048.

[9] J.R. Cash, *On the numerical integration of nonlinear two-point boundary value problems using iterated deferred corrections, Part* 2*: The development and analysis of highly stable deferred correction formulae*, SIAM J. Numer. Anal., 25 (1988), pp. 862–882.

[10] J.R. Cash and A. Singhal, *High order methods for the numerical solution of two-point boundary value problems*, BIT, 22 (1982), pp. 184–199.

[11] J.R. Cash and M.H. Wright, *A deferred correction method for nonlinear two-point boundary value problems: Implementation and numerical evaluation*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 971–989.

[12] W.H. Enright, K.R. Jackson, S.P. Nørsett, and P.G. Thomsen, *Interpolants for Runge–Kutta formulas*, ACM Trans. Math. Software, 12 (1986), pp. 193–218.

[13] W.H. Enright and P.H. Muir, *Efficient classes of Runge–Kutta methods for two-point boundary value problems*, Computing, 37 (1986), pp. 315–334.

[14] W.H. Enright and P.H. Muir, *Runge–Kutta software with defect control for boundary value ODEs*, SIAM J. Sci. Comput., 17 (1996), pp. 479–497.

[15] W.H. Enright and P.H. Muir, *Superconvergent interpolants for the collocation solution of BVODEs*, Tech. Report 309/97, Department of Computer Science, University of Toronto, Toronto, ON, 1997.

[16] S. Gupta, *An adaptive boundary value Runge–Kutta solver for first order boundary value problems*, SIAM J. Numer. Anal., 22 (1985), pp. 114–126.

[17] P.H. Muir, *Discrete and continuous mono-implicit Runge–Kutta schemes for BVODEs*, Adv. Comput. Math., 10 (1999), pp. 135–167.

[18] P.H. Muir and W.H. Enright, *Relationships among some classes of implicit Runge–Kutta methods and their stability functions*, BIT, 27 (1987), pp. 403–423.

[19] P. Muir and B. Owren, *Order barriers and characterizations for continuous mono-implicit Runge–Kutta schemes*, Math. Comp., 61 (1993), pp. 675–699.

[20] S. Pruess, *Interpolation schemes for collocation solutions of two-point boundary value problems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 322–333.

[21] S. Pruess and H. Jin, *A stable high-order interpolation scheme for superconvergent data*, SIAM J. Sci. Comput., 17 (1996), pp. 714–724.

[22] H. Stetter, *The Analysis of Discretization Methods for Ordinary Differential Equations*, Springer-Verlag, Berlin, 1982.

[23] J.H. Verner, *Differentiable interpolants for high-order Runge–Kutta methods*, SIAM J. Numer. Anal., 30 (1993), pp. 1446–1466.

[24] R. Weiss, *The application of implicit Runge–Kutta and collocation methods to boundary value problems*, Math. Comp., 28 (1974), pp. 449–464.