Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(4): 5634–5661.
DOI:10.3934/math.2022312
Received: 18 December 2019
Revised: 12 July 2021
Accepted: 18 July 2021
Published: 10 January 2022

Research article

Superconvergent interpolants for Gaussian collocation solutions of mixed order BVODE systems

M. Adams, J. Finden, P. Phoncharon and P. H. Muir[*]

Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia B3H 3C3, Canada

* **Correspondence:** Email: muir@smu.ca.

Abstract: The high quality COLSYS/COLNEW collocation software package is widely used for the numerical solution of boundary value ODEs (BVODEs), often through interfaces to computing environments such as Scilab, R, and Python. The continuous collocation solution returned by the code is much more accurate at a set of mesh points that partition the problem domain than it is elsewhere; the mesh point values are said to be superconvergent. In order to improve the accuracy of the continuous solution approximation at non-mesh points, when the BVODE is expressed in first order system form, an approach based on continuous Runge-Kutta (CRK) methods has been used to obtain a superconvergent interpolant (SCI) across the problem domain. Based on this approach, recent work has seen the development of a new, more efficient version of COLSYS/COLNEW that returns an error controlled SCI.
However, most systems of BVODEs include higher derivatives and a feature of COLSYS/COLNEW is that it can directly treat such mixed order BVODE systems, resulting in improved efficiency, continuity of the approximate solution, and user convenience. In this paper we generalize the approach mentioned above for first order systems to obtain SCIs for collocation solutions of mixed order BVODE systems. The main contribution of this paper is the derivation of generalizations of continuous Runge-Kutta-Nyström methods that form the basis for SCIs for this more general problem class. We provide numerical results that (i) show that the SCIs are much more accurate than the collocation solutions at non-mesh points, (ii) verify the order of accuracy of these SCIs, and (iii) show that the cost of utilizing the SCIs is a small fraction of the cost of computing the collocation solution upon which they are based.

Keywords: collocation; Runge-Kutta methods; Runge-Kutta-Nyström methods; boundary value ordinary differential equations; interpolation
Mathematics Subject Classification: 65L05, 65L10

1. Introduction

The numerical solution of boundary value ODEs (BVODEs) arises as a key task in a wide variety of applications such as the modelling of optimal control of electric vehicles [31], tropical fruit heat treatment [16], black holes [24], enzyme kinetics [30], and metal uptake in biointerphases [9].

The general form for a BVODE system with separated boundary conditions, written in first order form is,

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)), \quad \begin{pmatrix} \underline{g}_a(\underline{y}(a)) \\ \underline{g}_b(\underline{y}(b)) \end{pmatrix} = \underline{0}. \tag{1.1}$$

However, most BVODEs do not arise naturally in this form. It is common for higher derivatives to be present and in fact almost all of the problems considered in the applications cited above and in the well known BVODE collection in [4] include equations in which higher derivatives appear. These are referred to as mixed order systems. A standard form for a mixed order system consisting of n differential equations of orders $m_1, m_2, \ldots, m_n$, respectively, is

$$\begin{pmatrix} y_1^{(m_1)}(t) = f_1\left(t, \underline{z}(t)\right) \\ \cdots \\ y_n^{(m_n)}(t) = f_n\left(t, \underline{z}(t)\right) \end{pmatrix}, \quad \begin{pmatrix} \underline{g}_a(\underline{z}(a)) \\ \underline{g}_b(\underline{z}(b)) \end{pmatrix} = \underline{0}, \tag{1.2}$$

where

$$\underline{z}(t) = [y_1(t), y_1^{(1)}(t), \ldots, y_1^{(m_1-1)}(t), \quad \ldots, \quad y_n(t), y_n^{(1)}(t), \ldots, y_n^{(m_n-1)}(t)]^T, \tag{1.3}$$

where $y_j^{(\ell)}(t)$ is the ℓth derivative of the jth solution component, $y_j(t)$.

The high quality Fortran COLSYS/COLNEW collocation software package, [2, 3, 6] has been widely used for the numerical solution of BVODEs for several decades. More recently, interfaces to the package have been developed for several computing platforms such as Scilab, R, and Python. The papers cited at the beginning of this section are examples of investigations where COLSYS/COLNEW is used directly or through one of these platforms.

COLSYS/COLNEW is able to directly handle BVODEs having the more general mixed order form (1.2). While it is straightforward to convert a BVODE system of the form (1.2) to the form (1.1), *there are advantages to treating the mixed order form directly; these include user convenience, efficiency improvements, and higher continuity for some of the approximate solution components;* see, e.g., [4] and [26], for further discussion. See, in particular, [4], Page 252, where the authors comment that, for the application of collocation directly to the BVODE mixed order system form rather than the corresponding first order system form, “the saving in computation and storage is substantial”. See also, Table 7.3.1 of [12], where the computation times for COLSYS/COLNEW applied to a standard test problem are presented; the costs for the solution of the mixed order system form are approximately one third of those associated with the first order system form.

COLSYS/COLNEW employs an iterative, adaptive error control strategy in order to obtain a continuous solution approximation with an associated error estimate that satisfies a user-specified tolerance. The solver assumes that the solution components, $\{y_j(t)\}_{j=1}^n$, are approximated by piecewise polynomials of degree $k + m_j$, which are determined by imposing boundary and continuity conditions (the approximation to $y_j(t)$ will be required to have $C^{(m_j-1)}$-continuity), and by imposing collocation conditions that require the approximate solution to satisfy the BVODE at a set of k collocation points on each subinterval of a mesh that partitions the problem domain. The derivatives of the $\{y_j(t)\}_{j=1}^n$ solution components are approximated by the corresponding derivatives of the piecewise polynomials that approximate $\{y_j(t)\}_{j=1}^n$.

On each subinterval, the collocation points are the images of the set of k Gauss points mapped onto $[0, 1]$. It can be shown - see, e.g., [4] - that the $\underline{z}(t)$ values at the mesh points are substantially more accurate than elsewhere. Let $h = \max_{i=1}^{N} h_i$, where $h_i = t_{i+1} - t_i$, and where $\{t_i\}_{i=0}^N$ are the mesh points that partition the problem interval. Then, under appropriate assumptions [4], the global error associated with $z_j(t_i)$ is $O(h^{2k})$, *for all components of* $\underline{z}(t_i)$. On the other hand, at a non-mesh point, t, the global error associated with the component of $\underline{z}(t)$ equal to $y_j^{(\ell)}(t)$, $j = 1, \ldots, n$, $\ell = 0, \ldots, m_j - 1$, is $O(h^{k+m_j-\ell})$, provided that $k + m_j - \ell \leq 2k$. Therefore, when $2k > k + m_j - \ell$, the mesh point values are more accurate than the approximations at non-mesh points, and the former are said to be *superconvergent*.

An essential point is that COLSYS/COLNEW controls an error estimate for the continuous collocation solution rather than an error estimate for the mesh point solution values. Due to the lower accuracy of the collocation solution at non-mesh point values, it takes substantially more computational time and memory for COLSYS/COLNEW to obtain a satisfactory solution than it would if the continuous approximate solution had the same high accuracy that the mesh point values have. More mesh iterations are required to obtain a satisfactory continuous collocation solution and the meshes upon which the solution is constructed have to be much finer, which implies more subintervals, which implies higher computational and storage costs.

1.1. Superconvergent interpolants for collocation solutions

As explained above, a primary issue that impacts substantially on the efficiency of the computation performed by COLSYS/COLNEW is that the continuous collocation solution has substantially lower accuracy than that of the mesh point collocation solution values. A key idea in efforts to address this issue have focused on developing interpolants for *all* components of $\underline{z}(t)$ that have an error that is $O(h^{2k})$ (the mesh point error) for any t in the problem domain. Such interpolants are referred to as superconvergent interpolants (SCIs).

There have been a number of papers that have investigated the development of SCIs for collocation solutions. To our knowledge, the first such effort was [33], which considered the development of SCIs based on superconvergent mesh point values from several adjacent subintervals; difficulties are reported for highly non-uniform meshes. Subsequently, [34] and [20] considered SCIs based on *secondary collocation*; this involves performing an additional collocation computation, similar in cost to the original collocation computation, and efficiency is therefore a concern.

The paper, [13], describes how to develop SCIs *for first order BVODE systems*, (1.1), based on the use of continuous Runge-Kutta (CRK) methods - see, e.g., [28, 29, 36]. The key idea is to embed information from the collocation solution within a CRK method which is constructed to have the same superconvergent accuracy as the mesh point collocation values have. The primary contribution of that paper is the derivation of CRK methods that provide the basis for SCIs for collocation solutions for k, the number of collocation points, equal to 1, 2, 3, and 4, leading to SCIs of orders 2,4, 6, and 8, respectively.

In [13], the SCI is computed in *a post-processing step* to augment the collocation solution computed by COLSYS/COLNEW. That is, COLSYS/COLNEW is first called with a given user tolerance and the code returns with a continuous collocation solution that has an error estimate that satisfies the tolerance. Then the routine that implements the SCI is called to obtain a continuous approximate solution that has an error that is of the same order of accuracy as that of the mesh point values. *This means that the SCI is substantially more accurate than the user tolerance.* What of course is more desirable is to have the code return an SCI with a corresponding error estimate that meets the requested tolerance. This would mean that COLSYS/COLNEW would compute a continuous collocation solution with an intermediate level of accuracy and then the SCI would be used to augment this solution to obtain a continuous approximate whose accuracy would meet the requested tolerance. This would lead to a more efficient computation because COLSYS/COLNEW would only be required to compute a collocation solution of intermediate accuracy.

This idea has been pursued in recent work, [1], which has involved the development of COLNEWSC, a modification of COLSYS/COLNEW *that employs the SCI within the code, as the primary numerical solution that is returned to the user.* On a given mesh, COLNEWSC first computes the collocation solution, and then from this collocation solution, an SCI is constructed. An error estimate for this SCI is also computed and used within an adaptive error control algorithm to refine the mesh if necessary. The goal of this solver is to obtain an SCI for which the associated error estimate satisfies the user tolerance. In [1], COLNEWSC has been shown to be substantially more efficient than COLSYS/COLNEW since it is able to obtain a sufficiently accurate numerical solution using much coarser meshes than those employed by COLSYS/COLNEW due to the additional accuracy delivered by the SCI on a given mesh.

However, the algorithm upon which COLNEWSC is based requires that the BVODE be expressed as a first order system. *Thus, when COLSYS/COLNEW is applied directly to a mixed order BVODE system, the advantages associated with the introduction of an SCI are not available.* And, as mentioned above, in a majority of cases, BVODEs arise in their natural form as mixed order systems, which means that they can be treated directly by COLSYS/COLNEW, but not by COLNEWSC.

1.2. SCIs for mixed first/second order BVODE systems

In order to address this issue, the goal of this paper is to present a derivation of generalizations of CRK methods that can serve as the basis for SCIs for the most common subclass of (1.2) known as *mixed first and second order BVODE systems.* These systems have the form,

$$\underline{y}_1'(t) = \underline{f}_1\left(t, \underline{y}_1(t), \underline{y}_2(t), \underline{y}_2'(t)\right), \quad \underline{y}_2''(t) = \underline{f}_2\left(t, \underline{y}_1(t), \underline{y}_2(t), \underline{y}_2'(t)\right), \tag{1.4}$$

with boundary conditions

$$\begin{pmatrix} \underline{g}_a\left(\underline{y}_1(a), \underline{y}_2(a), \underline{y}_2'(a)\right) \\ \underline{g}_b\left(\underline{y}_1(b), \underline{y}_2(b), \underline{y}_2'(b)\right) \end{pmatrix} = \underline{0},$$

In this paper, we will derive schemes based on either Hermite interpolants or extensions of continuous Runge-Kutta-Nyström (CRKN) methods - see, e.g., [18, 23]. These schemes can provide the basis for SCIs for continuous approximations of $\underline{y}_1(t), \underline{y}_2(t)$, and $\underline{y}_2'(t)$, that will have the same superconvergent order of accuracy as the mesh point values. We will also consider preliminary implementations of these SCIs that are used in a post-processing step, similar to what was done in [13].

The computational costs associated with the construction and evaluation of an SCI for either a first or mixed order BVODE system are quite small compared to the costs associated with obtaining the collocation solution upon which the SCI is based. The primary costs associated with the computation of the collocation solution involve the setup and solution of a sequence of almost block diagonal linear systems performed within the context of a Newton type iteration. On the other hand, the costs associated with constructing an SCI involve only a few function evaluations per subinterval. Later in this paper, we provide numerical results similar to those given in [13], where it was shown that the cost of constructing an SCI is only a small fraction of the cost of the computation associated with obtaining the collocation solution upon which the SCI is based.

The significance of the work described in this paper is that it will provide the basis for an extension of COLNEWSC to allow for the direct treatment of mixed first and second order BVODE systems. This will mean that this new version of COLNEWSC will return an SCI for which the corresponding error estimate satisfies the user tolerance. Because the SCI is substantially more accurate than the collocation solution, this will mean that COLNEWSC will be able to terminate more quickly using a coarser mesh than that which is used by COLSYS/COLNEW. This new version of COLNEWSC will be able to directly handle mixed first and second order systems just as COLSYS/COLNEW can, thereby taking advantage of the user convenience and efficiency that is achieved by COLSYS/COLNEW for such systems. As well, the approach described in this paper suggests how the treatment of (1.4) can be generalized to allow for the treatment of general mixed order systems (1.2).

The schemes described in this paper may also be relevant for Gaussian collocation software for boundary value differential-algebraic systems [5], where the challenge will be to construct SCIs for the algebraic solution components. The schemes from this paper may also be applicable to software where Gaussian collocation is employed for the spatial descritization of PDEs - see, e.g., [1, 17, 22, 32] - that involve the second spatial derivative of the solution; it may be possible to adapt the schemes developed in this paper to obtain SCIs for the collocation solutions computed by these solvers.

Work that is related to the investigation discussed in this paper includes an alternative approach to developing SCIs based on *boot-strapping* [10]. The paper [14] describes the extension of this boot-strapping approach to general *mixed order* BVODE systems (1.2). Hermite-Birkhoff interpolants - see, e.g., [21] - are employed to provide SCIs for components of $\underline{z}(t)$ that approximate the solution components, $y_j(t), j = 1, \ldots, n$. Continuous approximations for the derivatives of $y_j(t), j = 1, \ldots, n$, are obtained by differentiating these SCIs; however, each differentiation leads to a decrease by one in the order of accuracy and therefore the approximation for a component of $\underline{z}(t)$ of the form $y_j^{(\ell)}(t)$ will have a global error that is only $O(h^{2k-\ell})$. Thus, in this approach, only the approximations to the $y_j(t)$ components will have the full superconvergent accuracy of the mesh point collocation solution values. As well, in [13] the boot-strapping approach is shown to require more evaluations of the right hand side of (1.1) than does the approach based on the use of CRK methods described in [13], which impacts negatively on the efficiency of these schemes. In more recent work, [35], the boot-strapping algorithm considered in [10] and [14] is extended to obtain SCIs for collocation solutions of Volterra integro-differential equations with delay.

This paper is organized as follows. We show, in Section 2, that it is possible to reformulate collocation methods, applied to (1.4), in a form that will allow us to show that they fit within the framework of Runge-Kutta (RK) and Runge-Kutta-Nyström (RKN) methods. This is followed, in Section 3, by a review of RKN and CRKN methods, Hermite interpolants, and a generalization of the CRKN methods, known as continuous parameterized implicit Runge-Kutta-Nyström (CPIRKN) methods. In Section 4, the main section of the paper, we derive specific CPIRKN methods that can provide the basis for SCIs of orders 2, 4, 6, and 8, for collocation solutions of (1.4). In Section 5, numerical results are provided to demonstrate the improved accuracy, orders of convergence, and low computational costs of the SCIs. Our summary, conclusions, and suggestions for future work are provided in Section 6.

2. Collocation methods in RK/RKN form

For the numerical solution of (1.4), and for the mesh, $\{t_i\}_{i=0}^N$, assume piecewise polynomials, i.e., collocation polynomials, $\underline{\hat{u}}_1(t)$ and $\underline{\hat{u}}_2(t)$, of degrees $k+1$ and $k+2$, respectively, which approximate $\underline{y}_1(t)$ and $\underline{y}_2(t)$, respectively. These collocation solutions must satisfy collocation conditions at k Gauss points per subinterval; on the ith subinterval, $[t_i, t_{i+1}]$, these conditions have the form,

$$\underline{\hat{u}}_1'(\hat{t}_r) = \underline{f}_1\left(\hat{t}_r, \underline{\hat{u}}_1(\hat{t}_r), \underline{\hat{u}}_2(\hat{t}_r), \underline{\hat{u}}_2'(\hat{t}_r)\right), \quad \underline{\hat{u}}_2''(\hat{t}_r) = \underline{f}_2\left(\hat{t}_r, \underline{\hat{u}}_1(\hat{t}_r), \underline{\hat{u}}_2(\hat{t}_r), \underline{\hat{u}}_2'(\hat{t}_r)\right), \tag{2.1}$$

where $r = 1, \ldots, k$, $\hat{t}_r = t_i + \rho_r h_i$, and ρ_r is the image of the rth Gauss point mapped onto $[0, 1]$. It is also required that $\underline{\hat{u}}_1(t)$ have C^0 continuity and that $\underline{\hat{u}}_2(t)$ have C^1 continuity.

The mesh point superconvergence results [4] imply that,

$$\left|\underline{y}_1(t_i) - \underline{\hat{u}}_1(t_i)\right| = O(h^{2k}), \quad \left|\underline{y}_2(t_i) - \underline{\hat{u}}_2(t_i)\right| = O(h^{2k}), \quad \left|\underline{y}_2'(t_i) - \underline{\hat{u}}_2'(t_i)\right| = O(h^{2k}), \tag{2.2}$$

and the non-mesh point convergence results [4] imply that,

$$\left|\underline{y}_1(t_i + \theta h_i) - \underline{\hat{u}}_1(t_i + \theta h_i)\right| = O(h^{k+1}), \quad \left|\underline{y}_2(t_i + \theta h_i) - \underline{\hat{u}}_2(t_i + \theta h_i)\right| = O(h^{k+2}),$$

$$\left|\underline{y}_2'(t_i + \theta h_i) - \underline{\hat{u}}_2'(t_i + \theta h_i)\right| = O(h^{k+1}), \tag{2.3}$$

for $\theta \in (0, 1)$. Therefore, for $k > 1$, the $\underline{\hat{u}}_1(t)$ and $\underline{\hat{u}}_2'(t)$ approximations at the mesh points will have a higher order of accuracy than elsewhere, and, for $k > 2$, all three of the $\underline{\hat{u}}_1(t)$, $\underline{\hat{u}}_2(t)$, and $\underline{\hat{u}}_2'(t)$ approximations at the mesh points will have a higher order of accuracy than elsewhere.

Let $\underline{y}_{1,i} = \underline{\hat{u}}_1(t_i)$, $\underline{y}_{2,i} = \underline{\hat{u}}_2(t_i)$, and $\underline{y}_{2,i}' = \underline{\hat{u}}_2'(t_i)$, and let $\underline{\hat{u}}_{1r} = \underline{\hat{u}}_1(\hat{t}_r)$, $\underline{\hat{u}}_{2r} = \underline{\hat{u}}_2(\hat{t}_r)$, and $\underline{\hat{u}}_{2r}' = \underline{\hat{u}}_2'(\hat{t}_r)$. Then, following from the collocation and continuity conditions, it can be shown that (see [27]), on the ith subinterval,

$$\underline{\hat{u}}_{1r} = \underline{y}_{1,i} + h_i\left(\hat{a}_{1,r1}\underline{\hat{f}}_{11} + \cdots + \hat{a}_{1,rk}\underline{\hat{f}}_{1k}\right), \tag{2.4}$$

$$\underline{\hat{u}}_{2r}' = \underline{y}_{2,i}' + h_i\left(\hat{a}_{2,r1}'\underline{\hat{f}}_{21} + \cdots + \hat{a}_{2,rk}'\underline{\hat{f}}_{2k}\right), \tag{2.5}$$

and

$$\underline{\hat{u}}_{2r} = \underline{y}_{2,i} + \rho_r h_i \underline{y}_{2,i}' + h_i^2\left(\hat{a}_{2,r1}\underline{\hat{f}}_{21} + \cdots + \hat{a}_{2,rk}\underline{\hat{f}}_{2k}\right), \tag{2.6}$$

where,

$$\underline{\hat{f}}_{1r} = \underline{f}_1\left(\hat{t}_r, \underline{\hat{u}}_{1r}, \underline{\hat{u}}_{2r}, \underline{\hat{u}}_{2r}'\right), \quad \underline{\hat{f}}_{2r} = \underline{f}_2\left(\hat{t}_r, \underline{\hat{u}}_{1r}, \underline{\hat{u}}_{2r}, \underline{\hat{u}}_{2r}'\right), \tag{2.7}$$

for $r = 1, \ldots, k$, and,

$$\hat{a}_{1,rj} = \hat{a}_{2,rj}' = \int_{\theta=0}^{\theta=\rho_r} \hat{b}_j(\theta)d\theta, \quad \text{and} \quad \hat{a}_{2,rj} = \int_{\tau=0}^{\tau=\rho_r}\left(\int_{\theta=0}^{\theta=\tau} \hat{b}_j(\theta)d\theta\right)d\tau,$$

for $r, j = 1, \ldots, k$, where $\{\hat{b}_j(\theta)\}_{j=1}^k$ are the Lagrange interpolating polynomials for the abscissa set $\{\rho_r\}_{r=1}^k$.

Similarly, it can be shown that,

$$\underline{y}_{1,i+1} = \underline{y}_{1,i} + h_i\left(\hat{b}_{11}\underline{\hat{f}}_{11} + \cdots + \hat{b}_{1k}\underline{\hat{f}}_{1k}\right), \tag{2.8}$$

$$\underline{y}'_{2,i+1} = \underline{y}'_{2,i} + h_i\left(\hat{b}'_{21}\underline{\hat{f}}_{21} + \cdots + \hat{b}'_{2k}\underline{f}_{2k}\right), \tag{2.9}$$

and

$$\underline{y}_{2,i+1} = \underline{y}_{2,i} + h_i\underline{y}'_{2,i} + h_i^2\left(\hat{b}_{21}\underline{\hat{f}}_{21} + \cdots + \hat{b}_{2k}\underline{\hat{f}}_{2k}\right), \tag{2.10}$$

where

$$\hat{b}_{1j} = \hat{b}'_{2j} = \int_{\theta=0}^{\theta=1} \hat{b}_j(\theta)d\theta, \quad \text{and} \quad \hat{b}_{2j} = \int_{\tau=0}^{\tau=1}\left(\int_{\theta=0}^{\theta=\tau} \hat{b}_j(\theta)d\theta\right)d\tau,$$

for $j = 1, \ldots, k$. *We note that the same collocation method is used for the approximation of* $\underline{y}_1(t)$ *and* $\underline{y}'_2(t)$.

3. CRKN methods, Hermite interpolants, and CPIRKN methods

3.1. Discrete and continuous RK and RKN methods

Assuming a first order ODE system, (1.1), and given solution approximations, $\underline{y}_i$, at the point t_i, and $\underline{y}_{i+1}$, at the point $t_{i+1} = t_i + h_i$, an s-stage RK method relates these solution approximations through the equation,

$$\underline{y}_{i+1} = \underline{y}_i + h_i\left(b_1\underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + b_s\underline{f}(\hat{t}_s, \underline{\hat{y}}_s)\right), \tag{3.1}$$

where $\hat{t}_r = t_i + c_r h_i$ and where

$$\underline{\hat{y}}_r = \underline{y}_i + h_i\left(a_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + a_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s)\right), \tag{3.2}$$

for $r = 1, \ldots, s$. The $\underline{f}(\hat{t}_1, \underline{\hat{y}}_1), \ldots, \underline{f}(\hat{t}_s, \underline{\hat{y}}_s)$ values are called the *stages* of the RK method. An RK method of a desired order of accuracy is determined by requiring that the coefficients of the method, $\{c_r, b_r, a_{r,j}\}_{r,j=1}^s$, satisfy a set of RK order conditions.

For the second order ODE system, $\underline{y}''(t) = \underline{f}(t, \underline{y}(t), \underline{y}'(t))$, an s-stage RKN method relates solution and derivative approximations, $\underline{y}_{i+1}$ and $\underline{y}'_{i+1}$, at t_{i+1}, to solution and derivative approximations, $\underline{y}_i$ and $\underline{y}'_i$, at t_i, through the formulas,

$$\underline{y}'_{i+1} = \underline{y}'_i + h_i\left(b'_1\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + b'_s\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right), \tag{3.3}$$

and

$$\underline{y}_{i+1} = \underline{y}_i + h_i\underline{y}'_i + h_i^2\left(b_1\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + b_s\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right), \tag{3.4}$$

where

$$\underline{\hat{y}}'_r = \underline{y}'_i + h_i\left(a'_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + a'_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right), \tag{3.5}$$

and

$$\underline{\hat{y}}_r = \underline{y}_i + c_r h_i\underline{y}'_i + h_i^2\left(a_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + a_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right), \tag{3.6}$$

for $r = 1, \ldots, s$. The $\underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1), \ldots, \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s)$ values are called the *stages* of the RKN method. An RKN method of a desired order of accuracy is determined by requiring that the coefficients of the method, $\{c_r, b_r, b'_r, a_{r,j}, a'_{rj}\}_{r,j=1}^{s}$, satisfy a set of RKN order conditions.

A comparison of the expressions for the collocation approximations given in the previous section with the general forms for the RK and RKN methods given above shows that the collocation methods are special cases of the RK and RKN methods. In particular, we note that the collocation formula, (2.8), (2.4), is a special case of the RK formula, (3.1), (3.2), and that the collocation formulas, (2.9),(2.5), (2.10), (2.6), are special cases of the RKN formulas, (3.3), (3.5), (3.4), (3.6).

A CRK method extends an RK method to provide a solution approximation at any point within the current subinterval. An s-stage CRK method approximates the solution, $\underline{y}(t)$, on the ith subinterval, $[t_i, t_{i+1}]$, by $\underline{u}(t)$ where, for $t \in [t_i, t_{i+1}]$, and $t = t_i + \theta h_i$,

$$\underline{u}(t_i + \theta h_i) = \underline{y}_i + h_i \left(b_1(\theta) \underline{f}(\hat{t}_1, \hat{\underline{y}}_1) + \cdots + b_s(\theta) \underline{f}(\hat{t}_s, \hat{\underline{y}}_s) \right), \tag{3.7}$$

where $\hat{\underline{y}}_r$ is defined as in (3.2) and $b_r(\theta)$ is a (weight) polynomial in θ. A CRK method of a desired order of accuracy is determined by requiring that the coefficients of the method, $\{c_r, b_r(\theta), a_{r,j}\}_{r,j=1}^{s}$, satisfy a set of continuous RK order conditions.

Similarly, CRKN methods, - see, e.g., [15,23,26], and references within - extend RKN methods to provide solution and derivative approximations at any point within the current subinterval. An s-stage CRKN method approximates the solution, $\underline{y}(t)$, on the ith step by $\underline{u}(t)$, where,

$$\underline{u}(t_i + \theta h_i) = \underline{y}_i + \theta h_i \underline{y}'_i + h_i^2 \left(b_1(\theta) \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + b_s(\theta) \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right), \tag{3.8}$$

and approximates the derivative, $\underline{y}'(t)$, by $\bar{\underline{u}}(t)$, where,

$$\bar{\underline{u}}(t_i + \theta h_i) = \underline{y}'_i + h_i \left(\bar{b}_1(\theta) \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + \bar{b}_s(\theta) \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right), \tag{3.9}$$

with $\hat{\underline{y}}_r$ and $\hat{\underline{y}}'_r$ defined as in (3.6) and (3.5), and where $b_r(\theta)$ and $\bar{b}_r(\theta)$ are (weight) polynomials in θ. A CRKN method of a desired order of accuracy is determined by requiring that the coefficients of the method, $\{c_r, b_r(\theta), b'_r(\theta), a_{r,j}, a'_{rj}\}_{r,j=1}^{s}$, satisfy a set of continuous RKN order conditions.

In this paper, we will employ CRK and CRKN methods in order to obtain SCIs for the collocation solutions of mixed first and second order systems (1.4). We will use a CRK method to handle the first order part of the system and a CRKN method to handle the second order part of the system.

In order to simplify the derivation of these methods, *we will require that the CRK and CRKN methods have the same abscissa, $\{c_r\}_{r=1}^{s}$, that the $\{a_{rj}\}_{r,j=1}^{s}$ coefficients of the CRK method be equal to the $\{a'_{rj}\}_{r,j=1}^{s}$ coefficients of the CRKN method, and that the $\{b_r(\theta)\}_{r=1}^{s}$ weight polynomials of the CRK methods be equal to the $\{\bar{b}_r(\theta)\}_{r=1}^{s}$ weight polynomials of the CRKN method.* We note that the collocation methods of the previous section are of this type.

Then, since the coefficients and weight polynomials used in the $\underline{y}_1(t)$ approximations will be the same as those used in the $\underline{y}'_2(t)$ approximations, there will be no need to provide the coefficients for the $\underline{y}_1(t)$ approximations separately; rather we can simply present the coefficients of the CRKN method with the understanding that the part of the CRKN formula that is used for the $\underline{y}'_2(t)$ approximation will also be used for the $\underline{y}_1(t)$ approximation.

Since COLSYS/COLNEW allows the user convenient access to the (already computed) collocation function evaluations, (2.7), the efficiency of the computation is improved when we embed these function evaluations within the CRKN methods we derive; i.e., *the collocation function evaluations will be employed as stages of the CRKN methods.* This is possible because, as we have seen, it is possible to represent a collocation method for a mixed first and second order system in terms of an RK method and an RKN method.

3.2. Hermite interpolants

For $k = 1$ and 2, it is possible to employ Hermite interpolants as the basis for SCIs of the desired orders of accuracy.

Once a collocation solution for (1.4) is obtained, we will have, for the ith subinterval, $[t_i, t_{i+1}]$, the six superconvergent data values,

$$\underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}, \quad \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}. \tag{3.10}$$

We can then compute the corresponding endpoint stages,

$$\underline{f}_{1,i} \equiv \underline{f}_1\left(t_i, \underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}\right), \quad \underline{f}_{1,i+1} \equiv \underline{f}_1\left(t_{i+1}, \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}\right), \tag{3.11}$$

and then construct a Hermite interpolant, $\underline{u}_1(t) \approx \underline{y}_1(t)$, having the form

$$\underline{u}_1(t_i + \theta h_i) = d_{10}(\theta)\underline{y}_{1,i} + d_{11}(\theta)\underline{y}_{1,i+1} + h_i\left(d_{12}(\theta)\underline{f}_{1,i} + d_{13}(\theta)\underline{f}_{1,i+1}\right), \tag{3.12}$$

where $d_{10}(\theta)$, $d_{11}(\theta)$, $d_{12}(\theta)$, and $d_{13}(\theta)$ are the Hermite cubic polynomials. They are defined to give $\underline{u}_1(t_i) = \underline{y}_{1,i}$, $\underline{u}_1(t_{i+1}) = \underline{y}_{1,i+1}$, $\underline{u}'_1(t_i) = \underline{f}_{1,i}$, $\underline{u}'_1(t_{i+1}) = \underline{f}_{1,i+1}$. The polynomials, $d_{10}(\theta)$ and $d_{11}(\theta)$, satisfy the condition that, $d_{10}(\theta) = 1 - d_{11}(\theta)$. This Hermite interpolant has an interpolation error that is $O(h^4)$.

Similarly, we can compute the endpoint stages,

$$\underline{f}_{2,i} \equiv \underline{f}_2\left(t_i, \underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}\right), \quad \underline{f}_{2,i+1} \equiv \underline{f}_2\left(t_{i+1}, \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}\right), \tag{3.13}$$

and form a Hermite interpolant, $\underline{u}_2(t) \approx \underline{y}_2(t)$, having the form,

$$\begin{aligned}\underline{u}_2(t_i + \theta h_i) = d_{20}(\theta)\underline{y}_{2,i} + d_{21}(\theta)\underline{y}_{2,i+1} + \\ h_i\left(d_{22}(\theta)\underline{y}'_{2,i} + d_{23}(\theta)\underline{y}'_{2,i+1}\right) + h_i^2\left(d_{24}(\theta)\underline{f}_{2,i} + d_{25}(\theta)\underline{f}_{2,i+1}\right),\end{aligned} \tag{3.14}$$

where $d_{20}(\theta), \ldots, d_{25}(\theta)$ are the Hermite quintic polynomials. They are defined to give $\underline{u}_2(t_i) = \underline{y}_{2,i}$, $\underline{u}_2(t_{i+1}) = \underline{y}_{2,i+1}$, $\underline{u}'_2(t_i) = \underline{y}'_{2,i}$, $\underline{u}'_2(t_{i+1}) = \underline{y}'_{2,i+1}$, $\underline{u}''_2(t_i) = \underline{f}_{2,i}$, $\underline{u}''_2(t_{i+1}) = \underline{f}_{2,i+1}$. The polynomials, $d_{10}(\theta)$, $d_{11}(\theta)$, $d_{22}(\theta)$, and $d_{23}(\theta)$, satisfy the conditions that, $d_{20}(\theta) = 1 - d_{21}(\theta)$ and that $d_{22}(\theta) = 1 - d_{23}(\theta)$.

We can also form a Hermite interpolant, $\underline{\bar{u}}_2(t) \approx \underline{y}'_2(t)$, of the form,

$$\underline{\bar{u}}_2(t_i + \theta h_i) = \bar{d}_{20}(\theta)\underline{y}'_{2,i} + \bar{d}_{21}(\theta)\underline{y}'_{2,i+1} + h_i\left(\bar{d}_{22}(\theta)\underline{f}_{2,i} + \bar{d}_{23}(\theta)\underline{f}_{2,i+1}\right), \tag{3.15}$$

where $\bar{d}_{20}(\theta)$, $\bar{d}_{21}(\theta)$, $\bar{d}_{22}(\theta)$, and $\bar{d}_{23}(\theta)$ are the Hermite cubic polynomials. They are defined to give $\underline{u}_2(t_i) = \underline{y}_{2,i}$, $\underline{u}_2(t_{i+1}) = \underline{y}_{2,i+1}$, $\underline{u}'_2(t_i) = \underline{f}_{2,i}$, $\underline{u}'_2(t_{i+1}) = \underline{f}_{2,i+1}$. The polynomials, $\bar{d}_{20}(\theta)$ and $\bar{d}_{21}(\theta)$, satisfy the condition that, $\bar{d}_{20}(\theta) = 1 - \bar{d}_{21}(\theta)$. The Hermite interpolant, (3.14), has an interpolation error of $O(h^6)$ and the Hermite interpolant, (3.15), has an interpolation error that is $O(h^4)$.

We note that the interpolation conditions will imply that $\underline{u}_1(t)$ and $\bar{\underline{u}}_2(t)$ will have C^1-continuity and that $\underline{u}_2(t)$ will have C^2-continuity.

When $k = 1$ or 2, the superconvergent collocation mesh point approximations have errors that are $O(h^2)$ and $O(h^4)$, respectively, and therefore the Hermite interpolants, (3.12), (3.14), (3.15), can be used to provide SCIs. For $k = 3$, the superconvergent collocation mesh point approximations have errors that are $O(h^6)$, and thus (3.14) can be used to provide an SCI for the $\underline{y}_2(t)$ approximation but (3.12) is not accurate enough to provide an SCI for $\underline{y}_1(t)$ nor is (3.15) accurate enough to provide one for $\underline{y}'_2(t)$. For $k = 4$, none of the Hermite interpolants are accurate enough to provide SCIs. We will consider these points further in the next section.

3.3. CPIRKN methods

For first order BVODE systems, (1.1), we have solution information associated with both endpoints of the ith subinterval, i.e., $\underline{y}_i$, $\underline{y}_{i+1}$. It is therefore useful to rewrite an RK method so that the stages have explicit dependence on the solution information from both endpoints. This leads to the methods known as parameterized implicit RK (PIRK) methods [11]; with the additional restriction that the each stage depend only on previously computed stages, we get the mono-implicit RK (MIRK) methods [7, 8]. By introducing weight polynomials, one can obtain continuous PIRK (CPIRK) methods; a subclass of these, the continuous MIRK methods, have been considered in [25].

For second order BVODE systems, $\underline{y}''(t) = \underline{f}(t, \underline{y}(t), \underline{y}'(t))$, we have solution and derivative information associated with the endpoints of the ith subinterval, i.e., $\underline{y}_i$, $\underline{y}'_i$, $\underline{y}_{i+1}$, $\underline{y}'_{i+1}$. It is possible to rewrite the well-known RKN methods so that the stages have explicit dependence on solution and derivative information at both endpoints of each subinterval; this gives the parameterized implicit RKN (PIRKN) methods. With the further restriction that each stage must depend only on previously computed stages, we get the mono-implicit RKN (MIRKN) methods; see, e.g., [26] and references within. By introducing weight polynomials, one can obtain the continuous PIRKN (CPIRKN) methods; a subclass of these, the continuous MIRKN methods, have been considered in [26].

The general form of a CPIRKN method is the same as that of a CRKN method, (3.8), and the definition of $\hat{t}_r$ is the same as in (3.2), but the $\underline{\hat{y}}_1, \ldots, \underline{\hat{y}}_s$ values and the $\underline{\hat{y}}'_1, \ldots, \underline{\hat{y}}'_s$ values are defined differently. Instead of the expressions for $\underline{\hat{y}}_r$ and $\underline{\hat{y}}'_r$ given in (3.6) and (3.5), we have

$$\underline{\hat{y}}_r = (1 - v_r)\underline{y}_i + v_r\underline{y}_{i+1} + h_i\left((c_r - v_r - w_r)\underline{y}'_i + w_r\underline{y}'_{i+1}\right) +$$

$$h_i^2\left(x_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + x_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right),$$

and

$$\underline{\hat{y}}'_r = (1 - v'_r)\underline{y}'_i + v'_r\underline{y}'_{i+1} + h_i\left(x'_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}}'_1) + \cdots + x'_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}}'_s)\right).$$

A CPRIKN method of a desired order of accuracy is determined by requiring that the coefficients of the method, $\{c_r, v_r, w_r, v'_r, b_r(\theta), b'_r(\theta), x_{r,j}, x'_{rj}\}_{r,j=1}^s$, satisfy a set of continuous order conditions. We represent the CPIRKN methods using a tableau of the form,

$$\begin{array}{c|c|c|cccc|c|cccc}
c_1 & v_1 & w_1 & x_{11} & x_{12} & \dots & x_{1s} & v'_1 & x'_{11} & x'_{12} & \dots & x'_{1s} \\
c_2 & v_2 & w_2 & x_{21} & x_{22} & \dots & x_{2s} & v'_2 & x'_{21} & x'_{22} & \dots & x'_{2s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
c_s & v_s & w_s & x_{s1} & x_{s2} & \dots & x_{ss} & v'_s & x'_{s1} & x'_{s2} & \dots & x'_{ss} \\
\hline
 & & & b_1(\theta) & b_2(\theta) & \dots & b_s(\theta) & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \dots & \bar{b}_s(\theta)
\end{array}. \tag{3.16}$$

It is possible to rewrite the Hermite interpolants, (3.12), (3.14), and (3.15), in terms of CPIRK and CPIRKN schemes. To see this, first recall that,

$$\underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}, \quad \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1},$$

satisfy (2.8), (2.9), and (2.10). We can therefore use these expressions to substitute for $\underline{y}_{1,i+1}$, $\underline{y}_{2,i+1}$, $\underline{y}'_{2,i+1}$ in (3.12), (3.14), and (3.15). These equations can then be rewritten, respectively, as

$$\underline{u}_1(t_i + \theta h_i) = \underline{y}_{1,i} + h_i \left(d_{12}(\theta)\underline{f}_{1,i} + d_{13}(\theta)\underline{f}_{1,i+1} + \hat{d}_{11}(\theta)\underline{\hat{f}}_{11} + \cdots + \hat{d}_{1k}(\theta)\underline{\hat{f}}_{1k} \right), \tag{3.17}$$

where $\hat{d}_{1j}(\theta) = \hat{b}_{1j} d_{11}(\theta)$,

$$\underline{\bar{u}}_2(t_i + \theta h_i) = \underline{y}'_{2,i} + h_i \left(\bar{d}_{22}(\theta)\underline{f}_{2,i} + \bar{d}_{23}(\theta)\underline{f}_{2,i+1} + \tilde{d}_{21}(\theta)\underline{\hat{f}}_{21} + \cdots + \tilde{d}_{2k}(\theta)\underline{\hat{f}}_{2k} \right), \tag{3.18}$$

where $\tilde{d}_{2j}(\theta) = \hat{b}'_{2j}\bar{d}_{21}(\theta)$, and

$$\underline{u}_2(t_i + \theta h_i) = \underline{y}_{2,i} + \theta h_i \underline{y}'_{2,i} + h_i^2 \left(d_{24}(\theta)\underline{f}_{2,i} + d_{25}(\theta)\underline{f}_{2,i+1} + \hat{d}_{21}(\theta)\underline{\hat{f}}_{21} + \cdots + \hat{d}_{2k}(\theta)\underline{\hat{f}}_{2k} \right), \tag{3.19}$$

where $\hat{d}_{2j}(\theta) = \hat{b}_{2j} d_{21}(\theta) + \hat{b}'_{2j} d_{23}(\theta)$. These new forms for the Hermite interpolants, i.e., (3.17), (3.18), (3.19), now fit into the standard forms for CPIRK and CPIRKN schemes.

3.4. CPIRKN methods for SCIs

In this subsection, we discuss the CPIRKN methods that we will use in order to develop SCIs for mixed first and second order systems when $k \geq 3$. (For $k = 1$ or 2, recall that the Hermite interpolants can be employed as the basis for the SCIs.)

On each subinterval, these CPIRKN methods will employ three types of stages:
(i) the first two stages will be the endpoint stages, (3.11), (3.13), which are easily computed from the corresponding collocation solution and derivative values,
(ii) the next k stages will be the collocation stages, (2.7), which are already available from the collocation computation, and,
(iii) the rest of the stages will be MIRKN stages; each such stage can depend on stages of types (i) and (ii), as well as on previously computed MIRKN stages. These new MIRKN stages will have the form,

$$\underline{\hat{f}}_{1r} \equiv \underline{f}_1\left(\hat{t}_r, \underline{\hat{y}}_{1r}, \underline{\hat{y}}_{2r}, \underline{\hat{y}}'_{2r}\right), \quad \underline{\hat{f}}_{2r} \equiv \underline{f}_2\left(\hat{t}_r, \underline{\hat{y}}_{1r}, \underline{\hat{y}}_{2r}, \underline{\hat{y}}'_{2r}\right), \quad r = k+1, \ldots, s,$$

where $\hat{t}_r = t_i + c_r h_i$, and where

$$\hat{\underline{y}}_{1r} = (1 - v'_r)\underline{y}_{1,i} + v'_r \underline{y}_{1,i+1} + h_i \left(x'_{r1} \underline{f}_{1,i} + x'_{r2} \underline{f}_{1,i+1} + \sum_{j=1}^{r-1} x'_{r,j+2} \hat{\underline{f}}_{1j} \right),$$

$$\hat{\underline{y}}_{2r} = (1 - v_r)\underline{y}_{2,i} + v_r \underline{y}_{2,i+1} + h_i \Big((c_r - v_r - w_r)\underline{y}'_{2,i} + w_r \underline{y}'_{2,i+1} \Big) +$$
$$h_i^2 \left(x_{r1} \underline{f}_{2,i} + x_{r2} \underline{f}_{2,i+1} + \sum_{j=1}^{r-1} x_{r,j+2} \hat{\underline{f}}_{2j} \right),$$

and

$$\hat{\underline{y}}'_{2r} = (1 - v'_r)\underline{y}_{2,i} + v'_r \underline{y}_{2,i+1} + h_i \left(x'_{r1} \underline{f}_{2,i} + x'_{r2} \underline{f}_{2,i+1} + \sum_{j=1}^{r-1} x'_{r,j+2} \hat{\underline{f}}_{2j} \right).$$

4. Derivation of CPIRKN methods for SCIs

4.1. Collocation with $k = 1$

When $k = 1$, the collocation mesh point values have errors that are $O(h^2)$. For this value of k, the continuous collocation solutions also have errors that are $O(h^2)$. We can therefore improve upon the continuous collocation solutions only in terms of achieving higher continuity, which we can do by employing the Hermite interpolants (3.12), (3.14), (3.15). Representing these interpolants in the form of a CPIRKN method, we get a CPIRKN method with the tableau:

$$\begin{array}{c|c|c|ccc|c|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & \frac{1}{2} \\
\hline
 & & & b_1(\theta) & b_2(\theta) & b_3(\theta) & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta)
\end{array}, \tag{4.1}$$

where

$$\begin{array}{c|c|c}
r & b_r(\theta) & \bar{b}_r(\theta) \\
\hline
1 & -\frac{1}{2}\theta^2(\theta-1)^3 & \theta(\theta-1)^2 \\
2 & \frac{1}{2}\theta^3(\theta-1)^2 & \theta^2(\theta-1) \\
3 & -\frac{1}{2}\theta^3(\theta-2) & -\theta^2(2\theta-3)
\end{array}. \tag{4.2}$$

This scheme, in addition to having the same order as the collocation approximations at the mesh points, provides approximations for $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ that are C^1-continuous, and an approximation for $\underline{y}_2(t)$ that is C^2-continuous (whereas the corresponding collocation polynomials have one order of continuity less in each case).

4.2. Collocation with $k = 2$

In this case, the mesh point collocation values have errors that are $O(h^4)$. The continuous collocation solution approximating $\underline{y}_2(t)$ will also have an error that is $O(h^4)$ but the continuous collocation solutions approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ will have errors that are only $O(h^3)$. The use of the Hermite interpolant, (3.14), allows us to improve the continuity of the $\underline{y}_2(t)$ approximation and we can use the Hermite interpolants, (3.12) and (3.15), to improve the order of accuracy and the continuity of the continuous approximations for $\underline{y}_1(t)$ and $\underline{y}'_2(t)$.

Expressing the Hermite interpolants in CPIRKN form gives a method with the tableau:

$$\begin{array}{c|c|c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\
\frac{1}{2}-\frac{\sqrt{3}}{6} & 0 & 0 & 0 & 0 & \frac{1}{36} & \frac{5}{36}-\frac{\sqrt{3}}{12} & \dots \\
\frac{1}{2}+\frac{\sqrt{3}}{6} & 0 & 0 & 0 & 0 & \frac{5}{36}+\frac{\sqrt{3}}{12} & \frac{1}{36} & \dots \\
\hline
 & & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & \dots
\end{array} \tag{4.3}$$

$$\begin{array}{c|c|cccc}
\dots & 0 & 0 & 0 & 0 & 0 \\
\dots & 1 & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\
\dots & 0 & 0 & 0 & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\
\hline
\dots & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta) & \bar{b}_4(\theta)
\end{array}, \tag{4.4}$$

where

r	$b_r(\theta)$	$\bar{b}_r(\theta)$
1	$-\frac{1}{2}\theta^2(\theta-1)^3$	$\theta(\theta-1)^2$
2	$\frac{1}{2}\theta^3(\theta-1)^2$	$\theta^2(\theta-1)$
3	$\frac{1}{12}\sqrt{3}\left(6\theta^2-15\theta-\theta\sqrt{3}+10+2\sqrt{3}\right)\theta^3$	$-\frac{1}{2}\theta^2(2\theta-3)$
4	$-\frac{1}{12}\sqrt{3}\left(6\theta^2-15\theta+\theta\sqrt{3}+10-2\sqrt{3}\right)\theta^3$	$-\frac{1}{2}\theta^2(2\theta-3)$

. (4.5)

4.3. Collocation with $k = 3$

In this case, the mesh point collocation values have errors that are $O(h^6)$, while the continuous collocation solution approximating $\underline{y}_2(t)$ has an error that is only $O(h^5)$ and the errors for the continuous collocation solutions approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ are only $O(h^4)$. While the use of the Hermite interpolant (3.14) can provide an SCI of the appropriate order for $\underline{y}_2(t)$, we cannot use the Hermite interpolants, (3.12) and (3.15), to obtain SCIs for $\underline{y}_1(t)$ and $\underline{y}'_2(t)$. We therefore consider the derivation of a CPIRKN scheme that will provide SCIs for $\underline{y}_1(t)$ and $\underline{y}'_2(t)$, with C^1-continuity, for which the errors are $O(h^6)$. Using the terminology associated with RK methods, this means that the CPIRKN must be of fifth order. (The definition of order for RK methods says that a pth order RK method has a local error that is $O(h^{p+1})$.)

As indicated earlier in this paper, the CPIRKN scheme will employ the two endpoint stages and the three collocation stages. For these five stages we observe that, for $q = 3$, the coefficients satisfy,

$$X'\underline{c}^j + \frac{\underline{v}'}{j+1} = \frac{\underline{c}^{j+1}}{j+1} \quad \text{for} \quad j = 0, 1, \dots, q, \tag{4.6}$$

and

$$X\underline{c}^{j-1} + \frac{\underline{v}}{j+1} + \frac{\underline{w}}{j} = \frac{\underline{c}^{j+1}}{j(j+1)} \quad \text{for} \quad j = 1, 2, \dots, q, \tag{4.7}$$

where X' is the s by s matrix whose r, jth element is x'_{rj}, X is the s by s matrix whose r, jth element is x_{rj}, $\underline{v}' = [v'_1, \ldots, v'_s]^T$, $\underline{v} = [v_1, \ldots, v_s]^T$, $\underline{w} = [w_1, \ldots, w_s]^T$, $\underline{c}^0$ is an s vector of 1's, and $\underline{c}^j = [c_1^j, \ldots, c_s^j]^T$, for $j > 0$. The above conditions, (4.6), (4.7), are called the *stage order conditions* up to order q - see [26].

A fifth order CPIRKN scheme satisfying these stage order conditions will have to satisfy the following order conditions involving the $\underline{\bar{b}}(\theta)$ polynomials:

$$\underline{\bar{b}}(\theta)^T \underline{c}^j = \frac{\theta^{j+1}}{j+1}, \quad \text{for} \quad j = 0, 1, 2, 3, 4,$$

and the single condition,

$$\underline{\bar{b}}(\theta)^T \left(X'\underline{c}^3 + \frac{\underline{v}'}{4} - \frac{\underline{c}^4}{4} \right) = 0,$$

where $\underline{\bar{b}}(\theta) = [\bar{b}_1(\theta), \ldots, \bar{b}_s(\theta)]^T$. Since there are six independent order conditions involving the $\underline{\bar{b}}(\theta)$ polynomials, this CPIRKN scheme will require a total of six stages. Since the scheme will use the two endpoint stages and the three collocation stages, one extra MIRKN stage will be required.

We will require that the new sixth stage have the maximum possible stage order, which turns out to be stage order six - that is, (4.6), (4.7), with $q = 6$. This forces $c_6 = \frac{1}{2} \pm \frac{\sqrt{10}}{10}$, with v_6, w_6, and v'_6 left free. We choose, arbitrarily, $c_6 = \frac{1}{2} - \frac{\sqrt{10}}{10}$, $v_6 = c_6$, $w_6 = -c_6$, and $v'_6 = c_6$. We can then solve the above order conditions to obtain the $\underline{\bar{b}}(\theta)$ weight polynomials.

As mentioned earlier, the approximation for $\underline{y}_2(t)$ can be determined using Hermite interpolation, with the Hermite interpolant converted to CPIRKN form, using the approach described in the previous section.

The resultant scheme has the tableau:

$$\begin{array}{c|c|c|cccccc|c}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
\frac{1}{2}-\frac{\sqrt{15}}{10} & 0 & 0 & 0 & 0 & \frac{1}{120} & \frac{1}{12}-\frac{\sqrt{15}}{45} & \frac{13}{120}-\frac{\sqrt{15}}{36} & 0 & \ldots \\
\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{5}{96}+\frac{\sqrt{15}}{72} & \frac{1}{48} & \frac{5}{96}-\frac{\sqrt{15}}{72} & 0 & \ldots \\
\frac{1}{2}+\frac{\sqrt{15}}{10} & 0 & 0 & 0 & 0 & \frac{13}{120}+\frac{\sqrt{15}}{36} & \frac{1}{12}+\frac{\sqrt{15}}{45} & \frac{1}{120} & 0 & \ldots \\
\frac{1}{2}-\frac{\sqrt{10}}{10} & v_6 & w_6 & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 0 & \ldots \\
\hline
 & & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta) & 0 & \ldots
\end{array}$$

$$\begin{array}{c|c|cccccc}
\ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\ldots & 0 & 0 & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} & 0 & 0 \\
\ldots & 0 & 0 & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} & 0 & 0 \\
\ldots & 0 & 0 & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} & 0 & 0 \\
\ldots & \frac{1}{2}-\frac{\sqrt{10}}{10} & x'_{61} & x'_{62} & x'_{63} & x'_{64} & x'_{65} & 0 \\
\hline
\ldots & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta) & \bar{b}_4(\theta) & \bar{b}_5(\theta) & \bar{b}_6(\theta)
\end{array}, \qquad (4.8)$$

where $v_6 = \frac{1}{2} - \frac{\sqrt{10}}{10}$, $w_6 = -\frac{1}{2} + \frac{\sqrt{10}}{10}$, and the values for x_{6r} and x'_{6r} for $r = 1, \ldots, 5$, the expressions for

$b_r(\theta)$, for $r = 1, \ldots, 5$, and the expressions for $\bar{b}_r(\theta)$, for $r = 1, \ldots, 6$, are given by,

r	x_{6r}	x'_{6r}
1	$\frac{27}{8000} + \frac{9}{8000}\sqrt{10}$	$\frac{3}{160} + \frac{3}{500}\sqrt{10}$
2	$\frac{27}{8000} - \frac{9}{8000}\sqrt{10}$	$-\frac{3}{160} + \frac{3}{500}\sqrt{10}$
3	$-\frac{29}{4800}\sqrt{6} + \frac{1709}{14400} - 1/36\sqrt{10}$	$\frac{1}{150}\sqrt{10} + \frac{3}{160}\sqrt{15}$
4	$\frac{407}{2250} - \frac{2}{45}\sqrt{10}$	$-\frac{19}{750}\sqrt{10}$
5	$\frac{29}{4800}\sqrt{6} + \frac{1709}{14400} - 1/36\sqrt{10}$	$\frac{1}{150}\sqrt{10} - \frac{3}{160}\sqrt{15}$

, (4.9)

r	$b_r(\theta)$
1	$-\frac{1}{2}\theta^2(\theta-1)^3$
2	$\frac{1}{2}\theta^3(\theta-1)^2$
3	$-\frac{1}{108}\sqrt{15}\left(-18\theta^2+45\theta+\theta\sqrt{15}-30-2\sqrt{15}\right)\theta^3$
4	$-\frac{2}{9}\theta^3(\theta-2)$
5	$-\frac{1}{108}\sqrt{15}\left(18\theta^2-45\theta+\theta\sqrt{15}+30-2\sqrt{15}\right)\theta^3$

, (4.10)

r	$\bar{b}_r(\theta)$
1	$-\frac{1}{18}\left(\sqrt{10}-1\right)\left(24\theta^2-7\theta+5\theta\sqrt{10}-2-2\sqrt{10}\right)\theta(\theta-1)^2$
2	$\frac{1}{18}\left(1+\sqrt{10}\right)\left(24\theta^2-41\theta+5\theta\sqrt{10}-3\sqrt{10}+15\right)(\theta-1)\theta^2$
3	$\frac{1}{18}\left(\sqrt{6}+1\right)(242\theta+3\sqrt{6}-63-3\theta^2\sqrt{15}-2\theta\sqrt{6}+6\theta\sqrt{15}-18\theta\sqrt{10}$ $+9\sqrt{10}-300\theta^2+120\theta^3+9\theta^2\sqrt{10}-3\sqrt{15})\theta^2$
4	$\frac{4}{9}\theta^2\left(-9+46\theta-60\theta^2+24\theta^3\right)$
5	$-\frac{1}{18}\left(\sqrt{6}-1\right)(242\theta-3\sqrt{6}+3\theta^2\sqrt{15}+2\theta\sqrt{6}-6\theta\sqrt{15}-18\theta\sqrt{10}$ $+9\sqrt{10}-300\theta^2+120\theta^3+9\theta^2\sqrt{10}+3\sqrt{15}-63)\theta^2$
6	$-\frac{40}{3}\theta^2(2\theta-1)(\theta-1)^2$

. (4.11)

4.4. *Collocation with* $k = 4$

When $k = 4$, the mesh point collocation values have errors that are $O(h^8)$. On the other hand, the continuous collocation solution approximating $\underline{y}_2(t)$ has an error that is only $O(h^6)$ and the errors for the continuous collocation solutions approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ are only $O(h^5)$. Since none of the Hermite interpolants can provide SCIs of the desired order, we will derive a CPIRKN scheme that will provide an SCI approximating $\underline{y}_2(t)$, with C^2-continuity, for which the error is $O(h^8)$, and SCIs approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$, with C^1-continuity, for which the errors are $O(h^8)$.

Since we want this CPIRKN scheme to have an error that is $O(h^8)$, the CPIRKN must be of seventh order. This CPIRKN scheme will employ the two endpoint stages and the four collocation stages. These six stages have coefficients that satisfy the stage order conditions, (4.6) and (4.7), for $q = 4$. The additional MIRKN stages will be required to satisfy the stage order conditions for $q = 6$.

A CPIRKN scheme satisfying the above stage order conditions will have to satisfy the following order conditions in order to achieve seventh order:

$$\underline{\bar{b}}(\theta)^T \underline{c}^j = \frac{\theta^{j+1}}{j+1}, \quad \text{for} \quad j = 0, 1, \ldots, 6, \quad \underline{b}(\theta)^T \underline{c}^j = \frac{\theta^{j+2}}{(j+1)(j+2)}, \quad \text{for} \quad j = 0, 1, \ldots, 5,$$

the three conditions involving $\underline{\bar{b}}(\theta)$,

$$\underline{\bar{b}}(\theta)^T \left(X' \left(X' \underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5} \right) \right) = 0, \quad \underline{\bar{b}}(\theta)^T \left(X' \underline{c}^5 + \frac{\underline{v}'}{6} - \frac{\underline{c}^6}{6} \right) = 0, \quad \underline{\bar{b}}(\theta)^T \left(\underline{c} \left(X' \underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5} \right) \right) = 0,$$

and the single condition involving $\underline{b}(\theta)$,

$$\underline{b}(\theta)^T \left(X' \underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5} \right) = 0,$$

where $\underline{b}(\theta) = [b_1(\theta), \ldots, b_s(\theta)]^T$.

Since there are a total of ten order conditions involving the $\underline{\bar{b}}(\theta)$ polynomials, it would appear then that this CPIRKN scheme will require a total of ten stages. However, it turns out that we can satisfy the order condition,

$$\underline{\bar{b}}(\theta)^T \left(X' \left(X' \underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5} \right) \right) = 0, \tag{4.12}$$

through a careful choice of the available free coefficients - see [27] for details. This allows us to reduce the number of order conditions from ten to nine which implies that the scheme will require only nine stages.

Since this scheme will include the two endpoint stages and four collocation stages, we will need three extra MIRKN stages. Applying the stage order conditions, (4.6), (4.7), with $q = 6$, to the MIRKN stages leads to the specification of most of the coefficients but there are several free coefficients that remain; we arbitrarily set $v_7 = c_7$, $v_8 = c_8$, $v_9 = c_9$, $w_7 = w_8 = w_9 = 0$, $x_{76} = x_{86} = x_{87} = x_{96} = x_{97} = x_{98} = 0$. The final step is to solve the nine remaining order conditions involving the $\underline{\bar{b}}(\theta)$ polynomials and the seven order conditions involving the $\underline{b}(\theta)$ polynomials in order to obtain the $\underline{\bar{b}}(\theta)$ and $\underline{b}(\theta)$ polynomials. Since there are only seven order conditions involving the $\underline{b}(\theta)$ polynomials, we can set $b_8(\theta) \equiv b_9(\theta) \equiv 0$ and use the seven order conditions involving the $\underline{b}(\theta)$ polynomials to determine $b_1(\theta), \ldots, b_7(\theta)$.

The resultant scheme has the tableau:

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

$$
\begin{array}{c|c|ccccccccc}
\dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\dots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & x'_{33} & x'_{34} & x'_{35} & x'_{36} & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & x'_{43} & x'_{44} & x'_{45} & x'_{46} & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & x'_{53} & x'_{54} & x'_{55} & x'_{56} & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & x'_{63} & x'_{64} & x'_{65} & x'_{66} & 0 & 0 & 0 & 0 \\
\dots & v'_7 & x'_{71} & x'_{72} & x'_{73} & x'_{74} & x'_{75} & x'_{76} & 0 & 0 & 0 \\
\dots & \frac{1}{5} & x'_{81} & x'_{82} & x'_{83} & x'_{84} & x'_{85} & x'_{86} & x'_{87} & 0 & 0 \\
\dots & \frac{4}{5} & x'_{91} & x'_{92} & x'_{93} & x'_{94} & x'_{95} & x'_{96} & x'_{97} & 0 & 0 \\
\hline
\dots & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta) & \bar{b}_4(\theta) & \bar{b}_5(\theta) & \bar{b}_6(\theta) & \bar{b}_7(\theta) & \bar{b}_8(\theta) & \bar{b}_9(\theta)
\end{array}
, \qquad (4.13)
$$

where $c_3 = \frac{1}{2} - \frac{1}{70}\sqrt{525 + 70\sqrt{30}}$, $c_4 = \frac{1}{2} - \frac{1}{70}\sqrt{525 - 70\sqrt{30}}$, $c_5 = \frac{1}{2} + \frac{1}{70}\sqrt{525 - 70\sqrt{30}}$, $c_6 = \frac{1}{2} + \frac{1}{70}\sqrt{525 + 70\sqrt{30}}$, and $c_7 = v_7 = \frac{1}{2} + 1/14\sqrt{7}$.

The expressions for the non-zero components of the X and X' matrices and the coefficients of the $\underline{b}(\theta)$ and $\underline{\bar{b}}(\theta)$ polynomials are too complicated to present exactly. Here we represent these values in terms of 20 digit decimal numbers.

The non-zeros of the matrix X are

$$
\begin{aligned}
[x_{33}, \dots, x_{36}] = [&.32305531606773849038 \times 10^{-2}, -.12501978957195100011 \times 10^{-2}, \\
&.59911990284166764 7 \times 10^{-3}, -.16908467308653538 \times 10^{-3}], \\
[x_{43}, \dots, x_{46}] = [&.44654739516219931749 \times 10^{-1}, .11055161125036900810 \times 10^{-1}, \\
&-.15763439131757701 42 \times 10^{-2}, .31957112533586327 71 \times 10^{-3}], \\
[x_{53}, \dots, x_{56}] = [&.10477319401975273714, .10928215124631229915, \\
&.11055161125036900810 \times 10^{-1}, -.666856745256879005 \times 10^{-3}], \\
[x_{63}, \dots, x_{66}] = [&.14960613448280714923, .19642483580315200379, \\
&.83717022845102756843 \times 10^{-1}, .32305531606773849038 \times 10^{-2}],
\end{aligned}
$$

$$[x_{71},\dots,x_{75}] = [.56407299162219992243\times10^{-2}, -.56407299162219991643\times10^{-2},$$
$$-.12818262090389426184\times10^{-1}, -.27667766089641676157\times10^{-1},$$
$$-.66656828962826040579\times10^{-1}],$$
$$[x_{81},\dots,x_{85}] = [.46987308407158757980\times10^{-2}, -.14560641740492091960\times10^{-2},$$
$$-.18777651201651205208\times10^{-1}, -.41033995097025767855\times10^{-1},$$
$$-.23431020367989693539\times10^{-1}],$$
$$[x_{91},\dots,x_{95}] = [.99888010024382092459\times10^{-2}, -.67461343357715427659\times10^{-2},$$
$$-.18777651201651205061\times10^{-1}, -.95324451125056381245\times10^{-2},$$
$$-.54932570352509823295\times10^{-1}].$$

The non-zeros of the matrix X' are

$$[x'_{33},\dots,x'_{36}] = [.86963711284363464343\times10^{-1}, -.26604180084998793304\times10^{-1},$$
$$.12627462689404724524\times10^{-1}, -.355514968579568315\times10^{-2}],$$
$$[x'_{43},\dots,x'_{46}] = [.18811811749986807165, .16303628871563653566,$$
$$-.27880428602470895218\times10^{-1}, .673550059453815551 2\times10^{-2}],$$
$$[x'_{53},\dots,x'_{56}] = [.16719192197418877317, .35395300603374396654,$$
$$.16303628871563653566, -.14190694931141142966\times10^{-1}],$$
$$[x'_{63},\dots,x'_{66}] = [.17748257225452261183, .31344511474186834680,$$
$$.35267675751627186462, .86963711284363464343\times10^{-1}],$$
$$[x'_{71},\dots,x'_{76}] = [.12595035543861662683\times10^{-1}, .27901157992841254519\times10^{-1},$$
$$.31424458236000028921\times10^{-1}, .12784556454961043482,$$
$$-.24867668578255783089\times10^{-1}, -.17489854774405759784],$$
$$[x'_{81},\dots,x'_{87}] = [-.41944590273292959284\times10^{-2}, -.16060145828848513267\times10^{-2},$$
$$.14012820352866098544, -.26472409697747613300\times10^{-1},$$
$$-.15474899161669444823, -.33179698061006715161\times10^{-1},$$
$$.80073369457001938508\times10^{-1}],$$
$$[x'_{91},\dots,x'_{97}] = [.73255409726707038846\times10^{-2}, .99139854171151489343\times10^{-2},$$
$$.21610386356930788521\times10^{-1}, .80525407473982583723\times10^{-1},$$
$$-.47751174444964251771\times10^{-1}, -.15169751523273691284,$$
$$.80073369457001939545\times10^{-1}].$$

The polynomials, $b_1(\theta), \ldots, b_7(\theta)$, are

$$b_1(\theta) = -1.4838054098817121549\,\theta^7 + 6.3599856012526592100\,\theta^6 - 10.677124344467704701\,\theta^5 + \\ 8.7095135247042803904\,\theta^4 - 3.4085693716075227451\,\theta^3 + 0.50000000000000000000\,\theta^2,$$

$$b_2(\theta) = 2.0717501456738433973\,\theta^7 - 6.0844588431917852229\,\theta^6 + 6.3228756555322952849\,\theta^5 - \\ 2.6793753641846084896\,\theta^4 + 0.36920840617025503042\,\theta^3,$$

$$b_3(\theta) = 2.4008645956962080391\,\theta^7 - 10.051375502307884471\,\theta^6 + 16.198264590214086232\,\theta^5 - \\ 12.157487596307555198\,\theta^4 + 3.7715852335674557048\,\theta^3,$$

$$b_4(\theta) = -1.5977758278295572808\,\theta^7 + 6.0738981481079401498\,\theta^6 - 8.3024639918680125568\,\theta^5 + \\ 4.4450128192755651476\,\theta^4 - 0.40020561139055488968\,\theta^3,$$

$$b_5(\theta) = -6.0321156342090800130\,\theta^7 + 21.594087470436269686\,\theta^6 - 28.922144091532793184\,\theta^5 + \\ 17.193739762616692926\,\theta^4 - 3.7259604661751969129\,\theta^3,$$

$$b_6(\theta) = -3.5901441705395393389\,\theta^7 + 10.917155179517231353\,\theta^6 - 11.659926172782139077\,\theta^5 + \\ 5.0666626066202185145\,\theta^4 - 0.72167134110935482879\,\theta^3,$$

$$b_7(\theta) = 8.2312263010898373512\,\theta^7 - 28.809292053814430705\,\theta^6 + 37.040518354904268003\,\theta^5 - \\ 20.578065752724593291\,\theta^4 + 4.1156131505449186412\,\theta^3,$$

and the polynomials, $\bar{b}_1(\theta), \ldots, \bar{b}_9(\theta)$, are

$$\bar{b}_1(\theta) = 118.21553917666580224\,\theta^7 - 450.21272045166368619\,\theta^6 + 677.90745020470247929\,\theta^5 - \\ 506.63265771593045627\,\theta^4 + 190.31615074107904624\,\theta^3 - 30.593761954853199618\,\theta^2 + \theta,$$

$$\bar{b}_2(\theta) = -60.923872509999145945\,\theta^7 + 249.69188711833036825\,\theta^6 - 401.42828353803587618\,\theta^5 + \\ 316.73682438259717901\,\theta^4 - 122.33698407441238897\,\theta^3 + 18.260428621519868725\,\theta^2,$$

$$\bar{b}_3(\theta) = -312.89027249245509522\,\theta^7 + 1182.4848315645309359\,\theta^6 - 1760.0136798995063288\,\theta^5 + \\ 1290.0528958373880041\,\theta^4 - 465.90057061724230909\,\theta^3 + 66.440723029853561290\,\theta^2,$$

$$\bar{b}_4(\theta) = -305.59222210601679616\,\theta^7 + 1134.2402328807602768\,\theta^6 - 1652.2852242034705354\,\theta^5 + \\ 1181.2977566097015877\,\theta^4 - 415.39180675681081626\,\theta^3 + 58.057336153267600487\,\theta^2,$$

$$\bar{b}_5(\theta) = 61.037789032562035884\,\theta^7 - 278.29971712366852415\,\theta^6 + 497.01008236447030656\,\theta^5 - \\ 432.96119140493011893\,\theta^4 + 182.24991389345060854\,\theta^3 - 28.710804184453033999\,\theta^2,$$

$$\bar{b}_6(\theta) = 182.44470556590992181\,\theta^7 - 725.92534732162277999\,\theta^6 + 1143.7888217385069610\,\theta^5 - \\ 890.88946104215982562\,\theta^4 + 341.54246348060257557\,\theta^3 - 50.787254998668143015\,\theta^2,$$

$$\bar{b}_7(\theta) = 214.91228070175434723\,\theta^7 - 752.19298245614029494\,\theta^6 + 1032.4385964912278336\,\theta^5 - \\ 700.61403508771909033\,\theta^4 + 238.12280701754382498\,\theta^3 - 32.666666666666657629\,\theta^2,$$

$$\bar{b}_8(\theta) = 429.13801384671252243\,\theta^7 - 1603.2561966116421409\,\theta^6 + 2349.7734629640512849\,\theta^5 - \\ 1685.4482584736159590\,\theta^4 + 588.91084526808341633\,\theta^3 - 79.117866993589184628\,\theta^2,$$

$$\bar{b}_9(\theta) = -326.34196121513359226\,\theta^7 + 1243.4700124011158454\,\theta^6 - 1887.1912261219461249\,\theta^5 + \\ 1428.4581268946686793\,\theta^4 - 537.51281895229395736\,\theta^3 + 79.117866993589188390\,\theta^2.$$

5. Numerical results

Here we present results for two test problems. Results for other test problems are given in [27].

Problem 1: The first problem consists of two nonlinear ODEs, one of third order, the other of second order; it has the form,

$$f'''(x) = \gamma^2 - 2f''(x)f(x) + (f'(x))^2 - g^2(x), \quad g''(x) = 2g(x)f'(x) - 2f(x)g'(x),$$

with boundary conditions

$$f'(0) = 0, \quad g(0) = 1, \quad f(0) = 0, \quad g(10) = \gamma, \quad f'(10) = 0.$$

We choose $\gamma = 3.0$.

In order to apply the methods developed in this paper, we will rewrite this problem as one first order ODE and two second order ODEs; letting $z_1(x) = f(x), z_2(x) = f'(x), z_3(x) = g(x)$, we get,

$$z_1'(x) = z_2(x), \quad z_2''(x) = \gamma^2 - 2z_2'(x)z_1(x) + z_2^2(x) - z_3^2(x),$$

$$z_3''(x) = 2z_3(x)z_2(x) - 2z_1(x)z_3'(x),$$

with the boundary conditions,

$$z_2(0) = 0, z_3(0) = 1, z_1(0) = 0, z_3(10) = \gamma, z_2(10) = 0.$$

This system has no known exact solution; in order to estimate the errors for the approximate solutions for this test problem, we computed a high-precision numerical solution to this problem using COLSYS/COLNEW.

Problem 2: The second problem consists of a coupled system of second order BVODEs obtained from the application of the transverse-method-of-lines with a fixed time step backward Euler method to discretize the time-dependent terms of the PDE:

$$z_t = z_{xx} - zz_x + \cos(\omega x) + t\omega^2\cos(\omega x) - t^2\cos(\omega x)\sin(\omega x), \tag{5.1}$$

where

$$z(0,t) = t, \quad z(1,t) = t\cos(\omega), \quad z(x,0) = 0, \quad \omega = 10, \quad t_{end} = 1.$$

The time step, Δt, is chosen to give a system of 20 second-order BVODEs; the ith BVODE has the form

$$z_i''(x) = \frac{z_{i+1}(x) - z_i(x)}{\Delta t} + z_i(x)z_i'(x) - \cos(\omega x) - t_i\omega^2\cos(\omega x) + t_i^2\cos(\omega x)\sin(\omega x),$$

where $z_i(x) \approx z(x, t_i)$ and $t_i = i \cdot \Delta t$. The corresponding boundary conditions are

$$z_i(0) = t_i, \quad z_i(1) = t_i\cos(\omega).$$

Because it is a second order system, the methods discussed in this paper can be directly applied to this problem.

For each test problem and for each solution component, the initial solution approximations provided to COLSYS/COLNEW are a straight line through the boundary conditions, for components with associated boundary conditions, and zero otherwise.

5.1. Convergence rates and maximum errors

In order to numerically verify the orders of convergence of the mesh point collocation values, the continuous collocation solutions, and the SCIs, we compute collocation solutions using COLSYS/COLNEW on a sequence of fixed uniform meshes. We consider meshes for which, N, the number of subintervals, is equal to 8, 16, 32, 64, and 128. (This required setting control parameters of COLSYS/COLNEW so that the mesh refinement capability was disabled.)

We provide numerical results for the cases $k = 3$ and 4, which are the ones where CPIRKN schemes are used to obtain the SCIs. We compute collocation solutions using COLSYS/COLNEW and then augment these to obtain the SCIs, based on the CPIRKN schemes derived earlier in this paper.

We consider numerical experiments in which we compute the maximum error *over all solution components* for a given evaluation point. The Mesh Point Solution error is the maximum error over all mesh point values. The errors for the Collocation Solution and for the Superconvergent Interpolant are the maximum errors of these continuous approximations over 10000 uniformly distributed sample points. Table 1 gives results for the $k = 3$ case, while Table 2 gives results for the $k = 4$ case.

Table 1. Error ratios (maximum errors), $k = 3$, Problem 1.

N	Mesh Point Solution	Collocation Solution	Superconvergent Interpolant
8	- (2.5x10^{-2})	- (4.0x10^{-2})	- (3.2x10^{-2})
16	52.0(4.8x10^{-4})	13.1(3.1x10^{-3})	51.8(6.2x10^{-4})
32	94.4(5.1x10^{-6})	12.0(2.6x10^{-4})	63.8(9.7x10^{-6})
64	59.0(8.6x10^{-8})	12.8(2.0x10^{-5})	62.5(1.6x10^{-7})
128	64.8(1.3x10^{-9})	14.0(1.4x10^{-6})	64.1(2.4x10^{-9})
Theoretical	64	16	64

Table 2. Error ratios (maximum errors), $k = 4$, Problem 1.

N	Mesh Point Solution	Collocation Solution	Superconvergent Interpolant
8	- (7.9x10^{-4})	- (6.1x10^{-3})	- (5.6x10^{-3})
16	124.8(6.4x10^{-6})	15.2(4.0x10^{-4})	191.2(2.9x10^{-5})
32	383.1(1.7x10^{-8})	24.6(1.6x10^{-5})	295.5(9.9x10^{-8})
64	277.5(6.0x10^{-11})	30.2(5.4x10^{-7})	219.3(4.5x10^{-10})
128	246.4(2.4x10^{-13})	32.1(1.7x10^{-8})	260.4(1.7x10^{-12})
Theoretical	256	32	256

The results demonstrate that the derived schemes achieve experimentally observable rates of convergence that are consistent with those predicted by the theory. We note that, for $k = 3$ and for the finest mesh, the error of the SCI is about three orders of magnitude smaller than that of the continuous collocation solutions. Similarly, for $k = 4$ and again for the finest mesh, the error of the SCI is about four orders of magnitude smaller than that of the continuous collocation solutions, for the finest mesh.

As well, we see that the error of the SCIs, for the finest meshes, is about 2 times larger, for $k = 3$, and about 7 times larger, for $k = 4$, than the error of the corresponding mesh point collocation solution values. There are several ways to improve the accuracy of the SCIs. We document these in the final section, as items for future work.

5.2. Relative costs of the collocation and SCI computations

Let n be the number of differential equations and k be the number of collocation points. A typical computation requires that several intermediate collocation solutions be computed on a sequence of meshes of sizes $N_1, N_2, \ldots, N_\ell$, where ℓ is the total number of meshes that are required. The total cost associated with obtaining the collocation solution is $O(N_1 q_1 (kn)^3 + N_2 q_2 (kn)^3 + \cdots + N_\ell q_\ell (kn)^3)$, where q_j is the number of Newton iterations required to obtain convergence for the computation of the jth intermediate collocation solution on the mesh of N_j subintervals. On the other hand, the cost associated with constructing an SCI in a post-processing step is $O(N_\ell n k^2)$. From these costs, we can see that the cost of constructing the SCI in a post-processing step, compared to the cost of computing the collocation solution, is quite small.

Table 3. Timing results in seconds, $k = 3$, Problem 2.

tol	COLSYS/COLNEW	SCI-SETUP	APPSLN	SCI-INTERP
10^{-2}	0.579	1.9×10^{-3}	1.3×10^{-3}	3.2×10^{-3}
10^{-3}	0.933	3.7×10^{-3}	1.3×10^{-3}	3.0×10^{-3}
10^{-4}	1.34	5.8×10^{-3}	1.1×10^{-3}	2.3×10^{-3}
10^{-5}	1.40	5.4×10^{-3}	1.1×10^{-3}	2.4×10^{-3}
10^{-6}	2.38	1.1×10^{-2}	1.0×10^{-3}	2.3×10^{-3}
10^{-7}	4.43	2.0×10^{-2}	1.1×10^{-3}	2.5×10^{-3}
10^{-8}	4.73	2.1×10^{-2}	1.0×10^{-3}	2.3×10^{-3}
10^{-9}	9.79	4.2×10^{-2}	9.9×10^{-4}	2.2×10^{-3}
10^{-10}	19.8	8.3×10^{-2}	9.4×10^{-4}	2.1×10^{-3}

Table 4. Timing results in seconds, $k = 4$, Problem 2.

tol	COLSYS/COLNEW	SCI-SETUP	APPSLN	SCI-INTERP
10^{-2}	0.532	1.2×10^{-3}	1.6×10^{-3}	3.2×10^{-3}
10^{-3}	0.866	1.9×10^{-3}	1.6×10^{-3}	3.3×10^{-3}
10^{-4}	1.33	2.9×10^{-3}	1.6×10^{-3}	3.2×10^{-3}
10^{-5}	1.27	2.7×10^{-3}	1.2×10^{-3}	2.4×10^{-3}
10^{-6}	2.53	5.6×10^{-3}	1.2×10^{-3}	2.4×10^{-3}
10^{-7}	3.14	8.2×10^{-3}	1.2×10^{-3}	2.4×10^{-3}
10^{-8}	4.45	1.5×10^{-2}	1.3×10^{-3}	2.4×10^{-3}
10^{-9}	7.74	2.2×10^{-2}	1.2×10^{-3}	2.4×10^{-3}
10^{-10}	7.48	2.3×10^{-2}	1.2×10^{-3}	2.4×10^{-3}

Tables 3 and 4 give timing results in seconds for Problem 2, with $\omega = 100$, for $k = 3$ and 4, and for a range of tolerances, $tol = 10^{-2}, \ldots, 10^{-10}$. We report the cost of the computation of the collocation solution with COLSYS/COLNEW and the cost of the call to the SCI-SETUP routine, which computes the extra stages that are needed for the SCI. We also report the cost of evaluating the collocation solution, using the COLSYS/COLNEW APPSLN routine, and the cost of evaluating the SCI, using the SCI-INTERP routine, at a 1000 uniformly spaced points across the problem domain. From these tables we see that the SCI setup cost is less than 0.5% of the cost of computing the collocation solution and that the cost of evaluating the SCI is about 2 to 3 times the cost of evaluating the collocation solution, with both of these costs being negligible compared to the cost of computing the collocation solution itself.

6. Summary, conclusions, and future work

In this paper we have shown how to generalize the approach considered in [13] to develop SCIs for Gaussian collocation solutions of mixed first and second order BVODE systems. These new methods can substantially improve the accuracy of the returned approximate solution leading to significant gains in the efficiency.

The approach involves augmenting the superconvergent collocation mesh point values with interpolants that have the same superconvergent order of accuracy. For the low order cases ($k = 1, 2$), Hermite interpolants can be employed on each subinterval to obtain the SCIs. For the $k = 3$ and 4 cases, a more general approach based on the use of a generalization of the continuous Runge-Kutta-Nyström methods can be employed. These methods can be used to obtain SCIs of orders 2, 4, 6, and 8, corresponding to $k = 1, 2, 3$, and 4. Thus the range of orders of SCIs provided in this paper is comparable to the range of orders provided by COLSYS/COLNEW.

Our numerical results show that
(i) it is possible to obtain SCIs that are substantially more accurate than the corresponding continuous collocation solutions,
(ii) the SCIs have experimentally observable orders of accuracy that are consistent with the expected orders, based on the theoretical framework that was used to derive them, and,
(iii) the computational cost associated with setting up an SCI is a small fraction of the cost associated with computing the collocation solution upon which the SCI is based.

As mentioned earlier, additional work could be done in order to improve the accuracy of the CPIRKN methods upon which the SCIs are based. One possibility could be to optimize of the choice of the free coefficients that arise during the derivations to attempt to minimize the magnitude of the leading order term in the error of each scheme. Another possibility could involve the use of CPIRKN methods that, for each k, are one order of accuracy higher than those derived in this paper. A third possibility could involve expressing the weight polynomials in a Barycentric Lagrange form rather than in terms of a monomial basis. The paper [19] shows that the use of a Barycentric Lagrange representation can lead to less interference from round-off error.

A major task for future work is to extend the work reported in [1] in order to incorporate the schemes described in this paper to allow COLNEWSC to directly treat mixed first and second order BVODE systems, leading to improved user convenience, numerical solutions with higher continuity, and a more efficient computation.

The collection of BVODEs given in [4] involves differential equations with solution derivatives of orders 1 to 4, and COLSYS/COLNEW is able to directly treat mixed order systems across this range of orders. This suggests another direction for future work which would involve extending the approach considered in this paper to derive methods leading to SCIs for mixed order systems involving derivatives of orders 1 to 4. The primary challenge will be to extend the CPIRKN methods to develop methods for differential equations in which derivatives of orders 3 and 4 appear. With this goal in mind, we have developed general forms for generalizations of CPIRKN methods for systems of differential equations involving derivatives of orders 3 and 4 - see the Appendix of this paper.

Finally, as mentioned earlier in this paper, because the boundary value differential-algebraic system solver COLDAE [5] is based on Gaussian collocation and the BACOL family of error control solvers for 1D PDEs (see [17, 32] and references within) makes use of Gaussian collocation for the spatial discretization of the PDEs, it may be worthwhile to investigate if it is possible to extend the ideas from the current paper to develop SCIs for the collocation solutions computed by these solvers.

Acknowledgments

(i) Funding sources: NSERC, Saint Mary's University.
(ii) The authors wish to thank the referees and the editor for many helpful comments.

Conflict of interest

All authors declare that they have no conflicts of interest in this paper.

References

1. M. Adams, C. Tannahill, P. H. Muir, Error control Gaussian collocation software for boundary value ODEs and 1D time-dependent PDEs, *Numer. Algorithms,* **81** (2019), 1505–1519. https://doi.org/10.1007/s11075-019-00738-2

2. U. M. Ascher, J. Christiansen, R. D. Russell, A collocation solver for mixed order systems of boundary value problems, *Math. Comp.,* **33** (1979), 659–679. https://doi.org/10.1090/S0025-5718-1979-0521281-7

3. U. M. Ascher, J. Christiansen, R. D. Russell, Collocation software for boundary value ODEs, *ACM Trans. Math. Softw.*, **7** (1981), 209–222. https://doi.org/10.1145/355945.355950

4. U. M. Ascher, R. M. M. Mattheij, R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics Series, Philadelphia: Society for Industrial and Applied Mathematics, 1995.

5. U. M. Ascher, R. J. Spiteri, Collocation software for boundary value differential-algebraic equations, *SIAM J. Sci. Comp.*, **15** (1994), 938–952. https://doi.org/10.1137/0915056

6. G. Bader, U. M. Ascher, A new basis implementation for a mixed order boundary value ODE solver, *SIAM J. Sci. Stat. Comp.*, **8** (1987), 483–500. https://doi.org/10.1137/0908047

7. K. Burrage, F. H. Chipman, P. H. Muir, Order results for mono-implicit Runge-Kutta methods, *SIAM J. Numer. Anal.*, **31** (1994), 876–891. https://doi.org/10.1137/0731047

8. J. R. Cash, A. Singhal, Mono-implicit Runge-Kutta formulae for the numerical integration of stiff differential systems, *IMA J. Numer. Anal.*, **2** (1982), 211–227. https://doi.org/10.1093/imanum/2.2.211

9. J. F. L. Duval, E. Rotureau, Dynamics of metal uptake by charged soft biointerphases: impacts of depletion, internalisation, adsorption and excretion, *Phys. Chem. Chem. Phys.*, **16** (2014), 7401–7416. https://doi.org/10.1039/C4CP00210E

10. W. H. Enright, K. R. Jackson, S. P. Nørsett, P. G. Thomsen, Interpolants for Runge-Kutta formulas, *ACM Trans. Math. Softw.*, **12** (1986), 193–218. https://doi.org/10.1145/7921.7923

11. W. H. Enright, P. H. Muir, Efficient classes of Runge-Kutta methods for two-point boundary value problems, *Computing*, **37** (1986), 315–334. https://doi.org/10.1007/BF02251090

12. W. H. Enright, P. H. Muir, *A Runge-Kutta Type Boundary Value ODE Solver with Defect Control*, Technical Report 93-267, Department of Computer Science, University of Toronto, Toronto, 1993.

13. W. H. Enright, P. H. Muir, Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations, *SIAM J. Sci. Comp.*, **21** (1999), 227–254. https://doi.org/10.1137/S1064827597329114

14. W. H. Enright, R. Sivasothinathan, Superconvergent interpolants for collocation methods applied to mixed order BVODEs, *ACM Trans. Math. Softw.*, **26** (2000), 323–351. https://doi.org/10.1145/358407.358410

15. J. M. Fine, Interpolants for Runge-Kutta-Nyström methods, *Computing*, **39** (1987), 27–42. https://doi.org/10.1007/BF02307711

16. A. D. Garnadi, P. D. R. Lestari, *Modeling hot water bath treatment of fruit using lateral method of lines in SCILAB*, Preprint 2020060054, 2020. https://doi.org/10.20944/preprints202006.0054.v1

17. K. R. Green, R. J. Spiteri, Extended BACOLI: Solving one-dimensional multiscale parabolic PDE systems with error control, *ACM Trans. Math. Softw.*, **45** (2019), 1–19. https://doi.org/10.1145/3301320

18. E. Hairer, S. P., Nörsett, G. Wanner, *Solving Ordinary Differential Equations. I. Nonstiff Problems*, Second edition, Springer Series in Computational Mathematics, 8, Berlin: Springer-Verlag, 1993.

19. N. J. Higham, The numerical stability of Barycentric Lagrange interpolation, *IMA J. Numer. Anal.*, **24** (2004), 547–556. https://doi.org/10.1093/imanum/24.4.547

20. H. Jin, S. Pruess, Uniformly superconvergent approximations for linear two-point boundary value problems, *SIAM J. Numer. Anal.*, **35** (1998), 363–375. https://doi.org/10.1137/S0036142996297205

21. S. Karlin, J. M. Karon, On Hermite-Birkhoff interpolation, *J. Approx. Theory*, **6** (1972), 90–115. https://doi.org/10.1016/0021-9045(72)90085-8

22. Z. Li, P. Muir, B-Spline Gaussian collocation software for two-dimensional parabolic PDEs, *Adv. Appl. Math. Mech.*, **5** (2013), 528–547. https://doi.org/10.4208/aamm.13-13S09

23. A. Marthinsen, Continuous extensions to Nyström methods for second order initial value problems, *BIT,* **36** (1996), 309–332. https://doi.org/10.1007/BF01731986

24. A. Marunovic, M. Murkovic, A novel black hole mimicker: a boson star and a global monopole nonminimally coupled to gravity, *Class. Quantum Grav.*, **31** (2014), 045010. https://doi.org/10.1088/0264-9381/31/4/045010

25. P. Muir, B. Owren, Order barriers and characterizations for continuous mono-implicit Runge-Kutta schemes, *Math. Comp.,* **61** (1993), 675–699. https://doi.org/10.1090/S0025-5718-1993-1195425-8

26. P. H. Muir, M. Adams, Mono-implicit Runge-Kutta-Nyström methods with application to boundary value ordinary differential equations, *BIT*, **41** (2001), 776–799.

27. P. H. Muir, M. Adams, J. Finden, P. Phoncharon, *Improving the Accuracy of Collocation Solutions of Mixed First and Second Order Boundary Value ODE Systems through the use of Superconvergent Interpolants*, Technical Report 2019_002, Department of Mathematics and Computing Science, Saint Mary's University, 2019.

28. B. Owren, M. Zennaro, Order barriers for continuous explicit Runge-Kutta methods, *Math. Comp.,* **56** (1991), 645–661. https://doi.org/10.1090/S0025-5718-1991-1068811-2

29. B. Owren, M. Zennaro, Derivation of optimal continuous explicit Runge-Kutta methods, *SIAM J. Sci. Stat. Comp.,* **13** (1992), 1488–1501. https://doi.org/10.1137/0913084

30. F. M. Pereira, S. C. Oliveira, Occurrence of dead core in catalytic particles containing immobilized enzymes: analysis for the Michaelis-Menten kinetics and assessment of numerical methods, *Bioprocess Biosyst. Eng.,* **39** (2016), 1717–1727. https://doi.org/10.1007/s00449-016-1647-0

31. N. Petit, A. Sciarretta, Optimal drive of electric vehicles using an inversion-based trajectory generation approach, *Proceedings of the 18th World Congress, The International Federation of Automatic Control, Milano, Italy*, (2011), 14519–14526. https://doi.org/10.3182/20110828-6-IT-1002.01986

32. J. Pew, Z. Li, C. Tannahill, P. Muir, G. Fairweather, Performance analysis of error-control B-spline Gaussian collocation software for PDEs, *Comput. Math. Appl.*, **77** (2019), 1888–1901. https://doi.org/10.1016/j.camwa.2018.11.025

33. S. Pruess, Interpolation schemes for collocation solutions of two point boundary value problems, *SIAM J. Sci. Stat. Comp.,* **7** (1986), 322–333. https://doi.org/10.1137/0907021

34. S. Pruess, H. Jin, A stable high-order interpolation scheme for superconvergent data, *SIAM J. Sci. Comp.*, **17** (1996), 714–724. https://doi.org/10.1137/S1064827593257481

35. M. Shakourifar, W. H. Enright, Superconvergent interpolants for collocation methods applied to Volterra integro-differential equations with delay, *BIT*, **52** (2012), 725–740. https://doi.org/10.1007/s10543-012-0373-5

36. J. H. Verner, Differentiable interpolants for high-order Runge-Kutta methods, *SIAM J. Numer. Anal.*, **30** (1993), 1446–1466. https://doi.org/10.1137/0730075

Appendix

A. Generalized CPIRK schemes for third order ODEs

In this subsection we give the general form for a generalized CPIRK method that can be applied directly to an ODE of the form,

$$y'''(t) = f\left(t, y(t), y'(t), y''(t)\right),$$

to provide SCIs for this problem class. The general form for these methods is,

$$\begin{aligned}
y_{i+1} &= y_i + \theta h_i y_i' + \frac{\theta^2 h_i^2}{2} y_i'' + h_i^3 \sum_{r=1}^{s} b_r(\theta) k_r,\\
y_{i+1}' &= y_i' + \theta h_i y_i'' + h_i^2 \sum_{r=1}^{s} \bar{b}_r(\theta) k_r,\\
y_{i+1}'' &= y_i'' + h_i \sum_{r=1}^{s} \tilde{b}_r(\theta) k_r,
\end{aligned}$$

where

$$k_r = f(\hat{t}_r, \hat{y}_r, \hat{y}_r', \hat{y}_r''), \quad \hat{t}_r = t_i + c_r h_i,$$

$$\hat{y}_r = (1 - v_r)y_i + v_r y_{i+1} + h_i\left((c_r - v_r - w_r)y_i' + w_r y_{i+1}'\right) +$$

$$h_i^2\left(\left(\frac{c_r^2}{2} - \frac{v_r}{2} - w_r - u_r\right)y_i'' + u_r y_{i+1}''\right) + h_i^3 \sum_{j=1}^{s} x_{rj} k_j,$$

$$\hat{y}_r' = (1 - v_r')y_i' + v_r' y_{i+1} + h_i\left((c_r - v_r' - w_r')y_i'' + w_r' y_{i+1}''\right) + h_i^2 \sum_{j=1}^{s} x_{rj}' k_j,$$

$$\hat{y}_r'' = (1 - v_r'')y_i'' + v_r'' y_{i+1}'' + h_i \sum_{j=1}^{s} x_{rj}'' k_j.$$

B. Generalized CPIRK schemes for fourth order ODEs

The methods presented in this subsection are generalizations of the methods presented in the previous subsection to allow for the direct treatment of BVODEs of the form,

$$y''''(t) = f(t, y(t), y'(t), y''(t), y'''(t)).$$

The general form for these methods is,

$$
\begin{aligned}
y_{i+1} &= y_i + \theta h_i y_i' + \frac{\theta^2 h_i^2}{2} y_i'' + \frac{\theta^3 h_i^3}{6} y_i''' + h_i^4 \sum_{r=1}^{s} b_r(\theta) k_r, \\
y_{i+1}' &= y_i' + \theta h_i y_i'' + \frac{\theta^2 h_i^2}{2} y_i''' + h_i^3 \sum_{r=1}^{s} \bar{b}_r(\theta) k_r, \\
y_{i+1}'' &= y_i'' + \theta h_i y_i''' + h_i^2 \sum_{r=1}^{s} \tilde{b}_r(\theta) k_r, \\
y_{i+1}''' &= y_i''' + h_i \sum_{r=1}^{s} \hat{b}_r(\theta) k_r,
\end{aligned}
$$

where

$$k_r = f(\hat{t}_r, \hat{y}_r, \hat{y}_r', \hat{y}_r'', \hat{y}_r'''), \quad \hat{t}_r = t_i + c_r h_i,$$

$$
\begin{aligned}
\hat{y}_r = {} & (1 - v_r) y_i + v_r y_{i+1} + h_i \left((c_r - v_r - w_r) y_i' + w_r y_{i+1}' \right) + \\
& h_i^2 \left(\left(\frac{c_r^2}{2} - \frac{v_r}{2} - w_r - u_r \right) y_i'' + u_r y_{i+1}'' \right) + \\
& h_i^3 \left(\left(\frac{c_r^3}{6} - \frac{v_r}{6} - \frac{w_r}{2} - u_r - z_r \right) y_i''' + z_r y_{i+1}''' \right) + h_i^4 \sum_{j=1}^{s} x_{rj} k_j,
\end{aligned}
$$

$$
\begin{aligned}
\hat{y}_r' = {} & (1 - v_r') y_{i-1}' + v_r' y_{i+1} + h_i \left((c_r - v_r' - w_r') y_i'' + w_r' y_{i+1}'' \right) + \\
& h_i^2 \left(\left(\frac{c_r^2}{2} - \frac{v_r'}{2} - w_r' - u_r' \right) y_i''' + u_r' y_{i+1}''' \right) + h_i^3 \sum_{j=1}^{s} x_{rj}' k_j,
\end{aligned}
$$

$$\hat{y}_r'' = (1 - v_r'') y_i'' + v_r'' y_{i+1}' + h_i \left((c_r - v_r'' - w_r'') y_i''' + w_r'' y_{i+1}''' \right) + h_i^2 \sum_{j=1}^{s} x_{rj}'' k_j,$$

$$\hat{y}_r''' = (1 - v_r''') y_i''' + v_r''' y_{i+1}''' + h_i \sum_{j=1}^{s} x_{rj}''' k_j.$$

AIMS Press