Superconvergent Interpolants for Collocation Solutions of Mixed First and Second Order Boundary Value ODE Systems

P.H. Muir, M.Adams, J. Finden, P. Phoncharon,*

Abstract

The high quality COLSYS/COLNEW Gaussian collocation software package has been widely used for the numerical solution of boundary value ordinary differential equations (BVODEs) for approximately four decades and, more recently, interfaces to the package have been developed for several high level problem solving environments/scripting languages such as Scilab, R, and Python. The collocation algorithm employs a mesh that partitions the domain of the BVODE and it is well-known that the computed continuous collocation solution has higher accuracy, i.e., is superconvergent, at the meshpoints. For systems of BVODEs represented as first order systems, it has been shown that an approach based on continuous Runge-Kutta (CRK) methods can be employed to obtain an efficiently implementable superconvergent interpolant (SCI) across the problem domain. Recent work has seen the development of a new version of COLNEW, called COLNEWSC, that computes and returns an error controlled SCI, leading to substantial efficiency improvements.

However, it is common for BVODEs to include higher derivatives and a feature of the COLSYS/COLNEW package is that it can directly handle mixed order BVODE systems. In this report we show that it is possible to extend the approach based on the use of CRK methods to obtain SCIs for collocation solutions of mixed first and second order BVODE systems. In addition to identifying a general framework for the study of these methods, based on continuous Runge-Kutta-Nyström methods, we derive specific methods that yield SCIs for mixed first and second order BVODE systems having orders of accuracy equal to 2, 4, 6, and 8. We also provide numerical results to verify the orders of convergence of the SCIs that we derive.

1 Introduction

Boundary value ordinary differential equations (BVODEs) arise in a wide variety of applications; the text [4] gives a number of examples. It is common for

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

software packages for the numerical solution of such problems to require that the user first convert the problem to a standard *first order system* form,

$$\underline{y}'(t) = \underline{f}(t, \underline{y}(t)). \tag{1}$$

However many BVODEs do not arise naturally in this form; it is common for higher derivatives to be present and in fact most of the problems in the BVODE collection in [4] include equations in which higher derivatives (up to and including order four) appear.

The widely used COLSYS [2, 3] and COLNEW [5] BVODE software packages are unusual in that they are able to directly treat mixed order systems. A standard form assumed by these packages for a mixed order system consisting of n differential equations of orders $m_1, m_2, \ldots, m_n$, respectively, is

$$\begin{array}{c} y_1^{(m_1)}(t) = f_1(t, \underline{z}(t)), \\ y_2^{(m_2)}(t) = f_2(t, \underline{z}(t)), \\ \vdots \\ y_n^{(m_n)}(t) = f_n(t, \underline{z}(t)), \end{array} \tag{2}$$

where

$$\begin{array}{rcl} \underline{z}(t) & = & [y_1(t), y_1^{(1)}(t), \ldots, y_1^{(m_1-1)}(t), \\ & & y_2(t), y_2^{(1)}(t), \ldots, y_2^{(m_2-1)}(t), \\ & & \vdots \\ & & y_n(t), y_n^{(1)}(t), \ldots, y_n^{(m_n-1)}(t)], \end{array} \tag{3}$$

where $y_j^{(\ell)}(t)$ is the ℓth derivative of the jth solution component, $y_j(t), j = 1, \ldots, n$.

While it is straightforward to convert a BVODE system of the form (2) to one of the form (1), there are advantages to treating the mixed order form directly; these include user convenience and efficiency improvements, as well as higher continuity for some of the approximate solution components; see, e.g., [4] and [20], for further discussion.

In this report we will focus on collocation methods for the numerical treatment of BVODEs; see, e.g., [4]. A collocation method for the numerical solution of (2) assumes that the solution components, $\{y_j(t)\}_{j=1}^n$, will be approximated by piecewise polynomials of degree $k + m_j$, which are determined by imposing continuity conditions (the approximation to $y_j(t)$ will be required to have $C^{(m_j-1)}$-continuity) and by requiring that the collocation solution satisfy the BVODE at a set of k collocation points within each subinterval of a mesh that partitions the problem interval. In this approach, the derivatives of the $\{\underline{y}_j(t)\}_{j=1}^n$ solution components are approximated by the corresponding derivatives of the piecewise polynomials that approximate $\{\underline{y}_j(t)\}_{j=1}^n$.

For certain choices of the collocation points, it can be shown - see, e.g., [4] - that the $\underline{z}(t)$ values at the mesh points are substantially more accurate than elsewhere [4]. Assume that on each subinterval the collocation points are the

images of the set of k Gauss points. Let $h = \max_{i=1,\ldots,N} h_i$, 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-meshpoint, 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 $k + m_j - \ell \leq 2k$. When $2k > k + m_j - \ell$, the discrete meshpoint values are more accurate than the approximations at non-mesh points and the meshpoint values are therefore said to be *superconvergent*.

In order to attempt to address the issue of the continuous collocation solution having lower accuracy than the mesh point collocation solution values, a key idea has been is to investigate how one might develop interpolants to augment the high accuracy collocation meshpoint values in order to obtain *continuous* approximations to the components of $\underline{z}(t)$ that have the same order of accuracy *over the whole problem interval* that the collocation values have at the mesh points. That is, the goal has been to improve upon the collocation polynomial at non-meshpoints by developing interpolants for *all* components of $\underline{z}(t)$ that have an error that is $O(h^{2k})$ for any t in the problem domain. Such interpolants are thus referred to as *superconvergent interpolants* (SCIs).

There have been a number of papers in which the question of developing SCIs for collocation solutions has been considered. To our knowledge, the first such effort was [25], which considers the development of SCIs based on superconvergent mesh values from several adjacent subintervals; difficulties are reported for highly non-uniform meshes. Subsequently, [26] and [17] consider SCIs based on *secondary collocation*; this involves performing an additional collocation computation, similar to the original collocation computation, and efficiency is therefore a concern.

The extension of the standard implicit Runge-Kutta (RK) methods, e.g., [7], to continuous RK (CRK) methods - see, e.g., [22, 23, 27] - has been discussed in the literature for several decades. The paper, [12], describes how to develop efficiently computable SCIs *for first order BVODE systems*, (1), based on the use of CRK methods. Since the approach we will employ in this report to develop SCIs for mixed first and second order BVODE systems is based on a generalization of the work reported in [12], we will discuss the main ideas from this paper in further detail later in this report.

Also, included in [12] is a brief description of an alternative approach to developing SCIs based on *boot-strapping* [9]. The paper [13] describes the extension of a boot-strapping approach to general *mixed order* BVODE systems (2). Hermite-Birkhoff interpolants - see, e.g., [18] - are employed to provide SCIs for components of $\underline{z}(t)$ that approximate $y_j(t), j = 1, \ldots, n$. Continuous approximations for the derivatives of $y_j(t), j = 1, \ldots, n$, are obtained by differentiating these SCIs; 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 $O(h^{2k-\ell})$; that is, only the approximations to the $y_j(t)$ components will have the full superconvergence of the mesh point

values. As well, the boot-strapping approach is shown in [12] to require more evaluations of the right hand side of (1) than does the approach based on the use of CRK methods described in [12].

It is important to note that COLSYS/COLNEW employs an error control algorithm when computing a collocation solution to a given BVODE; this means that the code employs an adaptive framework in order to eventually obtain a continuous solution approximation with an associated error estimate that satisfies a user-specified tolerance. Assuming that the BVODE being solved is of the form (1), the CRK approach [12] is applied in *a post-processing step* to augment the collocation solution returned by COLSYS/COLNEW in order to obtain the more accurate SCI. *This means that the SCI delivers more accuracy than the user has requested.* (This is because the collocation solution computed by COLSYS/COLNEW has an error estimate that satisfies the tolerance; the extra accuracy delivered by the SCI is beyond what the user actually asked for.) In order to remedy this situation, recent work [1], has involved the development of COLNEWSC, a modification of COLSYS/COLNEW that employs the SCI *as the primary numerical solution that is returned to the user.* On a given mesh, COLNEWSC first computes the collocation solution in the usual way, 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 obtain an SCI for which an associated error estimate satisfies the user tolerance. COLNEWSC is 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 COLNEW due to the additional accuracy delivered by the SCI on a given mesh.

The original developers of COLSYS/COLNEW might have chosen to only return an error controlled *meshpoint* collocation solution. This would allow the code to complete its computation much more quickly on a coarser mesh but it would also have meant sacrificing the return of the continuous approximate solution that is naturally available from the collocation computation because it would have much less accuracy than the user had requested. However, at least for first order BVODE systems, it would then be possible to augment the superconvergent discrete meshpoint collocation solution with an SCI, in a post-processing step, as in [12], thus recovering the capability of returning a continuous solution. In this case however, this modified version of COLSYS/COLNEW would not actually be controlling the error of the SCI; error control would only be applied to the meshpoint collocation solution. There would however be the expectation that, because it is of the same order as the discrete solution, the SCI would also have comparable accuracy. This is similar to the approach that is typically used in codes that compute numerical solutions to initial value ODEs. Error control is typically only applied to the discrete numerical solution at the end of a given step, and then a continuous numerical solution (based on some type of interpolant) of the same order of accuracy as the discrete numerical solution is constructed.

The work we describe in this report involves generalizing the approach from [12] to handle *mixed first and second order* BVODE systems. These systems

have the form

$$\underline{y}_1'(t) = \underline{f}_1\left(t, \underline{y}_1(t), \underline{y}_2(t), \underline{y}_2'(t)\right),$$
$$\underline{y}_2''(t) = \underline{f}_2\left(t, \underline{y}_1(t), \underline{y}_2(t), \underline{y}_2'(t)\right), \tag{4}$$

where

$$y_1 : R \rightarrow R^{n_1}, \quad y_2 : R \rightarrow R^{n_2}, \quad f_1 : R \times R^{n_1} \times R^{n_2} \times R^{n_2} \rightarrow R^{n_1},$$

and

$$f_2 : R \times R^{n_1} \times R^{n_2} \times R^{n_2} \rightarrow R^{n_2}.$$

We will obtain SCIs for the approximation of $\underline{y}_1(t), \underline{y}_2(t)$, and $\underline{y}_2'(t)$ across the entire problem domain that have an errors that are $O(h^{2k})$. That is, the continuous solution approximations that we obtain will have the same super-convergent accuracy as the meshpoint collocation values. When the number of collocation points, k, is 1 or 2, we will show that corresponding SCIs can be obtained from standard Hermite interpolants. For larger k values, the SCIs will be based on a subclass of the continuous Runge-Kutta-Nyström (CRKN) methods - see, e.g., [19],[15].

This report is organized as follows. We begin, in Section 2, with a review of the application of collocation to (4); we also show how to write collocation methods in a form that will allow us to view these methods as a subclass of the Runge-Kutta-Nyström (RKN) methods - see, e.g., [15]. This is followed, in Section 3, by a review of RKN and CRKN methods, Hermite interpolants, and parameterized CRKN methods. In Section 4, we derive specific Hermite interpolants and CRKN methods that can provide the basis for the SCIs for (4), for $k = 1, 2, 3$, and 4. In Section 5, we briefly describe the software we have developed that implements a CRKN-type SCI based on a given collocation solution for (4). In Section 6, numerical results are provided to demonstrate the improved accuracy and the order of convergence of the SCIs for the $k = 3$ and $k = 4$ cases. Our summary, conclusions, and suggestions for future work are provide in the final section.

2 Collocation methods for mixed first and second order systems

In this section we review collocation methods applied to (4) and derive expressions which will later be used to demonstrate the connection between collocation methods applied to (4) and the well-known RKN methods.

For the numerical solution of (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. Because they are collocation solutions, $\underline{\hat{u}}_1(t)$ and $\underline{\hat{u}}_2(t)$ must satisfy collocation conditions; on the ith subinterval, $[t_i, t_{i+1}]$, these conditions have the form,

$$\underline{\hat{u}}_1'(\hat{t}_{ir}) = \underline{f}_1\left(\hat{t}_{ir}, \underline{\hat{u}}_1(\hat{t}_{ir}), \underline{\hat{u}}_2(\hat{t}_{ir}), \underline{\hat{u}}_2'(\hat{t}_{ir})\right),$$

$$\underline{\hat{u}}_2''(\hat{t}_{ir}) = \underline{f}_2\left(\hat{t}_{ir}, \underline{\hat{u}}_1(\hat{t}_{ir}), \underline{\hat{u}}_2(\hat{t}_{ir}), \underline{\hat{u}}_2'(\hat{t}_{ir})\right), \tag{5}$$

where $r = 1, \ldots, k$, $\hat{t}_{ir} = t_i + \rho_r h_i$, and where ρ_r is the image of the rth Gauss point on $[0, 1]$. (A collocation method can employ other sets of points on each subinterval but the use of the Gauss points leads to the collocation solution having the superconvergence properties that are central to approach discussed in this report.) It is usually required that $\underline{\hat{u}}_1(t)$ have C^0 continuity and that $\underline{\hat{u}}_2(t)$ have C^1 continuity, leading to continuity conditions which must also be applied to the collocation polynomials. The collocation and continuity conditions, together with the boundary conditions, form a nonlinear system in which the unknowns are the coefficients of the piecewise polynomial basis functions in terms of which $\underline{\hat{u}}_1(t)$ and $\underline{\hat{u}}_2(t)$ are represented. In COLSYS/COLNEW, a collocation solution with an estimated global error satisfying a user-provided tolerance is computed in an iterative fashion, involving the solution of several nonlinear systems associated with a sequence of meshes which are chosen adaptively to approximately equidistribute the estimated error over the problem domain.

From [4], we have the meshpoint superconvergence results,

$$\left|\underline{y}_1(t_i) - \underline{\hat{u}}_1(t_i)\right| = O(h^{2k}), \left|\underline{y}_2(t_i) - \underline{\hat{u}}_2(t_i)\right| = O(h^{2k}), \left|\underline{y}_2'(t_i) - \underline{\hat{u}}_2'(t_i)\right| = O(h^{2k}), \tag{6}$$

and the non-mesh point convergence results,

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

$$\left|\underline{y}_2'(t_i + \theta h_i) - \underline{\hat{u}}_2'(t_i + \theta h_i)\right| = O(h^{k+1}), \tag{7}$$

for $\theta \in (0, 1)$. These results imply that, 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 that, for $k > 2$, the $\underline{\hat{u}}_2(t)$ approximations at the mesh points will have a higher order of accuracy than elsewhere.

We now discuss how to rewrite the collocation method for (4) so that a connection to RKN methods can be made apparent. See [28] where the corresponding connection between collocation methods for first order systems and RK methods is discussed; see also [4].

Once $\underline{\hat{u}}_1(t)$ and $\underline{\hat{u}}_2(t)$ are obtained, the collocation conditions (5) imply that, on the ith subinterval, $\underline{\hat{u}}_1'(t)$ and $\underline{\hat{u}}_2''(t)$ will satisfy,

$$\underline{\hat{u}}_1'(t_i + \theta h_i) = \hat{b}_1(\theta)\underline{\hat{f}}_{11} + \cdots + \hat{b}_k(\theta)\underline{\hat{f}}_{1k},$$

$$\underline{\hat{u}}_2''(t_i + \theta h_i) = \hat{b}_1(\theta)\underline{\hat{f}}_{21} + \cdots + \hat{b}_k(\theta)\underline{\hat{f}}_{2k}, \tag{8}$$

where, for $t \in [t_i, t_{i+1}]$, $t = t_i + \theta h_i$, with $\theta \in [0, 1]$,

$$\underline{\hat{f}}_{1r} = \underline{f}_1\left(\hat{t}_{ir}, \underline{\hat{u}}_1(\hat{t}_{ir}), \underline{\hat{u}}_2(\hat{t}_{ir}), \underline{\hat{u}}_2'(\hat{t}_{ir})\right), \quad \underline{\hat{f}}_{2r} = \underline{f}_2\left(\hat{t}_{ir}, \underline{\hat{u}}_1(\hat{t}_{ir}), \underline{\hat{u}}_2(\hat{t}_{ir}), \underline{\hat{u}}_2'(\hat{t}_{ir})\right), \tag{9}$$

$r = 1, \ldots, k$, and $\{\hat{b}_r(\theta)\}_{r=1}^k$ are the standard Lagrange interpolating polynomials for the abscissa set $\{\rho_r\}_{r=1}^k$. Integrating the equations that appear in (8) gives

$$\underline{\hat{u}}_1(t_i + \theta h_i) = \underline{y}_{1,i} + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \right) \underline{\hat{f}}_{11} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \right) \underline{\hat{f}}_{1k} \right), \quad (10)$$

$$\underline{\hat{u}}_2'(t_i + \theta h_i) = \underline{y}_{2,i}' + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \right) \underline{\hat{f}}_{2k} \right), \quad (11)$$

where $\underline{y}_{1,i}$ is the meshpoint collocation approximation for $\underline{y}_1(t_i)$ and $\underline{y}_{2,i}'$ is the meshpoint collocation approximation for $\underline{y}_2'(t_i)$. Integrating (11) gives

$$\underline{\hat{u}}_2(t_i + \theta h_i) = \underline{y}_{2,i} + \theta h_i \underline{y}_{2,i}' +$$

$$h_i^2 \left(\left(\int \int \hat{b}_1(\theta) d^2\theta \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \int \hat{b}_k(\theta) d^2\theta \right) \underline{\hat{f}}_{2k} \right), \quad (12)$$

where $\underline{y}_{2,i}$ is the meshpoint collocation approximation for $\underline{y}_2(t_i)$. From (10), (11), let

$$\underline{\hat{u}}_{1r} \equiv \underline{\hat{u}}_1(\hat{t}_{ir}) = \underline{y}_{1,i} + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{11} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{1k} \right), \quad (13)$$

$$\underline{\hat{u}}_{2r}' \equiv \underline{\hat{u}}_2'(\hat{t}_{ir}) = \underline{y}_{2,i}' + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{2k} \right), \quad (14)$$

and from (12), let

$$\underline{\hat{u}}_{2r} \equiv \underline{\hat{u}}_2(\hat{t}_{ir}) = \underline{y}_{2,i} + \rho_r h_i \underline{y}_{2,i}' +$$

$$h_i^2 \left(\left(\int \int \hat{b}_1(\theta) d^2\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \int \hat{b}_k(\theta) d^2\theta \Big|_0^{\rho_r} \right) \underline{\hat{f}}_{2k} \right). \quad (15)$$

As well, due to the continuity conditions, evaluation of (10), (11), and (12), with $\theta = 1$ gives,

$$\underline{y}_{1,i+1} = \underline{\hat{u}}_1(t_{i+1}) = \underline{y}_{1,i} + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \Big|_0^1 \right) \underline{\hat{f}}_{11} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \Big|_0^1 \right) \underline{\hat{f}}_{1k} \right), \quad (16)$$

$$\underline{y}_{2,i+1}' = \underline{\hat{u}}_2'(t_{i+1}) = \underline{y}_{2,i}' + h_i \left(\left(\int \hat{b}_1(\theta) d\theta \Big|_0^1 \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \hat{b}_k(\theta) d\theta \Big|_0^1 \right) \underline{\hat{f}}_{2k} \right), \quad (17)$$

and

$$\underline{y}_{2,i+1} = \underline{\hat{u}}_2(t_{i+1}) = \underline{y}_{2,i} + h_i \underline{y}_{2,i}' +$$

$$h_i^2 \left(\left(\int \int \hat{b}_1(\theta) d^2\theta \Big|_0^1 \right) \underline{\hat{f}}_{21} + \cdots + \left(\int \int \hat{b}_k(\theta) d^2\theta \Big|_0^1 \right) \underline{\hat{f}}_{2k} \right). \quad (18)$$

We can then rewrite (13), (14), (15), as,

$$\underline{\hat{u}}_{1r} = \underline{y}_{1,i} + h_i \left(\hat{a}_{1,r1} \underline{\hat{f}}_{11} + \cdots + \hat{a}_{1,rk} \underline{\hat{f}}_{1k} \right), \tag{19}$$

$$\underline{\hat{u}}'_{2r} = \underline{y}'_{2,i} + h_i \left(\hat{a}'_{2,r1} \underline{\hat{f}}_{21} + \cdots + \hat{a}'_{2,rk} \underline{\hat{f}}_{2k} \right), \tag{20}$$

and

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

where

$$\hat{a}_{1,rj} = \hat{a}'_{2,rj} = \int \hat{b}_j(\theta) d\theta \Big|_0^{\rho_r} \quad \text{and} \quad \hat{a}_{2,rj} = \int \int \hat{b}_j(\theta) d^2\theta \Big|_0^{\rho_r}.$$

Similarly, we can rewrite (16), (17), (18), as

$$\underline{y}_{1,i+1} = \underline{y}_{1,i} + h_i \left(\hat{b}_{11} \underline{\hat{f}}_{11} + \cdots + \hat{b}_{1k} \underline{\hat{f}}_{1k} \right), \tag{22}$$

$$\underline{y}'_{2,i+1} = \underline{y}'_{2,i} + h_i \left(\hat{b}'_{21} \underline{\hat{f}}_{21} + \cdots + \hat{b}'_{2k} \underline{f}_{2k} \right), \tag{23}$$

and

$$\underline{y}_{2,i+1} = \underline{y}_{2,i} + h_i \underline{y}'_{2,i} + h_i^2 \left(\hat{b}_{21} \underline{\hat{f}}_{21} + \cdots + \hat{b}_{2k} \underline{\hat{f}}_{2k} \right), \tag{24}$$

where

$$\hat{b}_{1j} = \hat{b}'_{2j} = \int \hat{b}_j(\theta) d\theta \Big|_0^{1} \quad \text{and} \quad \hat{b}_{2j} = \int \int \hat{b}_j(\theta) d^2\theta \Big|_0^{1}.$$

We note that the collocation method used for the approximation of $\underline{y}_1(t)$ is the same method that is used for the approximation of $\underline{y}'_2(t)$.

In the next section, we will relate the above form of the above collocation methods, (19) - (24), to the more general RK and RKN methods.

3 Runge-Kutta methods for mixed first and second order systems

3.1 Discrete and continuous Runge-Kutta and Runge-Kutta-Nyström methods

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

$$\underline{y}_{i+1} = \underline{y}_i + h_i \left(b_1 \underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + b_s \underline{f}(\hat{t}_s, \underline{\hat{y}}_s) \right), \tag{25}$$

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

$$\underline{\hat{y}}_r = \underline{y}_i + h_i \left(a_{r1} \underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + a_{rs} \underline{f}(\hat{t}_s, \underline{\hat{y}}_s) \right). \tag{26}$$

The $\underline{f}(\hat{t}_s, \hat{\underline{y}}_s), \ldots, \underline{f}(\hat{t}_s, \hat{\underline{y}}_s)$ values are called the *stages* of the RK method. The coefficients of the RK method are usually presented in the form of a tableau,

$$\begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ c_2 & a_{21} & \ldots & a_{2s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array} . \tag{27}$$

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,

$$\underline{y}'_{i+1} = \underline{y}'_i + h_i \left(b'_1 \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + b'_s \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right), \tag{28}$$

and

$$\underline{y}_{i+1} = \underline{y}_i + h_i \underline{y}'_i + h_i^2 \left(b_1 \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + b_s \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right), \tag{29}$$

where

$$\hat{\underline{y}}'_r = \underline{y}'_i + h_i \left(a'_{r1} \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + a'_{rs} \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right), \tag{30}$$

and

$$\hat{\underline{y}}_r = \underline{y}_i + c_r h_i \underline{y}'_i + h_i^2 \left(a_{r1} \underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1) + \cdots + a_{rs} \underline{f}(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s) \right). \tag{31}$$

The $\underline{f}(\hat{t}_1, \hat{\underline{y}}_1, \hat{\underline{y}}'_1), \ldots, f(\hat{t}_s, \hat{\underline{y}}_s, \hat{\underline{y}}'_s)$ values are called the *stages* of the RKN method. The coefficients of the RKN method are often presented in a tableau of the form,

$$\begin{array}{c|ccc|ccc} c_1 & a_{11} & \ldots & a_{1s} & a'_{11} & \ldots & a'_{1s} \\ c_2 & a_{21} & \ldots & a_{2s} & a'_{21} & \ldots & a'_{2s} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} & a'_{s1} & \ldots & a'_{ss} \\ \hline & b_1 & \ldots & b_s & b'_1 & \ldots & b'_s \end{array} . \tag{32}$$

An RK or RKN method of a desired order of accuracy is obtained by applying corresponding *order conditions* in order determine the coefficients of the method; see, e.g., [7].

A comparison of the expressions for the collocation approximations given in Section 2 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, (22), (19), is a special case of the RK formula, (25), (26), and that the collocation formulas, (23),(20), (24), (21), are special cases of the RKN formulas, (28), (30), (29), (31).

As mentioned earlier in this report, it is possible, by using a CRK method, to extend the standard (discrete) 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,

$$\underline{u}(t) = \underline{u}(t_i + \theta h_i) = \underline{y}_i + h_i \left(b_1(\theta)\underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + b_s(\theta)\underline{f}(\hat{t}_s, \underline{\hat{y}}_s) \right), \qquad (33)$$

where $\hat{t}_r$ and $\underline{\hat{y}}_r$ are defined as in (26) and $b_r(\theta)$ is a (weight) polynomial in θ.

Similarly, CRKN methods, extend discrete RKN methods to provide a solution and a derivative approximation 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, in this case,

$$\underline{u}(t) = \underline{u}(t_i+\theta h_i) = \underline{y}_i+\theta h_i \underline{y}'_i+h_i^2 \left(b_1(\theta)\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}'}_1) + \cdots + b_s(\theta)\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}'}_s) \right), \qquad (34)$$

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

$$\underline{\bar{u}}(t) = \underline{\bar{u}}(t_i+\theta h_i) = \underline{y}'_i+h_i \left(\bar{b}_1(\theta)\underline{f}(\hat{t}_1, \underline{\hat{y}}_1, \underline{\hat{y}'}_1) + \cdots + \bar{b}_s(\theta)\underline{f}(\hat{t}_s, \underline{\hat{y}}_s, \underline{\hat{y}'}_s) \right), \qquad (35)$$

with $\hat{t}_r$ defined in (26), $\underline{\hat{y}}_r$ and $\underline{\hat{y}'}_r$ defined as in (31) and (30), and where $b_r(\theta)$ and $\bar{b}_r(\theta)$ are (weight) polynomials in θ. (The number of stages, s, employed within a CRK or CRKN scheme is typically greater than the number of stages employed in the corresponding discrete scheme.)

The CRK and CRKN methods, respectively, are usually represented by the following tableaus which contain their coefficients and weight polynomials,

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ c_2 & a_{21} & \dots & a_{2s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1(\theta) & \dots & b_s(\theta) \end{array}, \qquad (36)$$

and

$$\begin{array}{c|ccc|ccc} c_1 & a_{11} & \dots & a_{1s} & a'_{11} & \dots & a'_{1s} \\ c_2 & a_{21} & \dots & a_{2s} & a'_{21} & \dots & a'_{2s} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} & a'_{s1} & \dots & a'_{ss} \\ \hline & b_1(\theta) & \dots & b_s(\theta) & \bar{b}_1(\theta) & \dots & \bar{b}_s(\theta) \end{array}. \qquad (37)$$

CRK and CRKN methods of a desired order of accuracy are obtained by requiring that the coefficients and weight polynomials satisfy corresponding continuous versions of the order conditions; see, e.g., [7].

In this report, as mentioned earlier, we will employ the framework of CRK and CRKN methods in order to obtain SCIs for the collocation solutions of mixed first and second order systems (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 method and the CRKN method share some coefficients*

and weight polynomials. To be more specific, both methods will have the same abscissa, $\{c_r\}_{r=1}^s$, the $\{a_{rj}\}_{r,j=1}^s$ coefficients of the CRK method will equal the $\{a'_{rj}\}_{r,j=1}^s$ coefficients of the CRKN method, and the $\{b_r(\theta)\}_{r=1}^s$ weight polynomials of the CRK methods will be equal to the $\{\bar{b}_r(\theta)\}_{r=1}^s$ weight polynomials of the CRKN method. Thus, we will have $\underline{y}_1(t) \approx \underline{u}_1(t), \underline{y}'_2(t) \approx \underline{\bar{u}_2}(t)$, and $\underline{y}_2(t) \approx \underline{u}_2(t)$, where, on the ith subinterval,

$$\underline{u}_1(t_i + \theta h_i) = \underline{y}_{1,i} + h_i \left(\bar{b}_1(\theta)\underline{f}_{11} + \cdots + \bar{b}_s(\theta)\underline{f}_{1s} \right),$$

$$\underline{\bar{u}_2}(t_i + \theta h_i) = \underline{y}'_{2,i} + h_i \left(\bar{b}_1(\theta)\underline{f}_{21} + \cdots + \bar{b}_s(\theta)\underline{f}_{2s} \right),$$

$$\underline{u}_2(t_i + \theta h_i) = \underline{y}_{2,i} + \theta h_i \underline{y}'_{2,i} + h_i^2 \left(b_1(\theta)\underline{f}_{21} + \cdots + b_s(\theta)\underline{f}_{2s} \right),$$

where

$$f_{1r} \equiv f_1 \left(\hat{t}_r, \underline{\hat{y}}_{1r}, \underline{\hat{y}}_{2r}, \underline{\hat{y}}'_{2r} \right), \quad f_{2r} \equiv f_2 \left(\hat{t}_r, \underline{\hat{y}}_{1r}, \underline{\hat{y}}_{2r}, \underline{\hat{y}}'_{2r} \right),$$

and where

$$\underline{\hat{y}}_{1r} = \underline{y}_{1,i} + h_i \left(a'_{r1} f_{11} + \cdots + a'_{rs} f_{1s} \right),$$

$$\underline{\hat{y}}_{2r} = \underline{y}_{2,i} + c_r h_i \underline{y}'_{2,i} + h_i^2 \left(a_{r1} f_{21} + \cdots + a_{rs} f_{2s} \right),$$

$$\underline{\hat{y}}'_{2r} = \underline{y}'_{2,i} + h_i \left(a'_{r1} f_{21} + \cdots + a'_{rs} f_{2s} \right).$$

Note: Since the coefficients and weight polynomials used in the $\underline{y}_1(t)$ approximations are the same as those used in the $\underline{y}'_2(t)$ approximations, there is no need to list the tableau of coefficients for the $\underline{y}_1(t)$ approximations separately; rather we can simply present the tableau of the CRKN method (37). A hybrid CRK/CRKN scheme for the treatment of a mixed first/second order system (4) will be given in terms of the CRKN formula 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, (9), it improves efficiency to embed these function evaluations within the CRK/CRKN methods we derive; i.e., the collocation function evaluations will be employed as stages of the CRK/CRKN methods. This will mean that some of the coefficients of our CRK/CRKN methods will be specified by the coefficients of the embedded collocation method. The specific way in which this is done will be described in more detail in the next section, when we derive specific CRK/CRKN methods of this type.

3.2 Hermite interpolants leading to SCIs

For the lower order cases, $k = 1$ and 2, it is possible to use standard Hermite interpolating polynomials on each subinterval in order to obtain a suitable SCI. We will now briefly review the details.

Once a collocation solution for (4) is obtained, we will have, for the ith subinterval, $[t_i, t_{i+1}]$, the six superconvergent data values,

$$\underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}, \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}. \tag{38}$$

We can then compute the corresponding endpoint stages,

$$\tilde{f}_{1,i} \equiv f_1\left(t_i, \underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}\right), \quad \tilde{f}_{1,i+1} \equiv f_1\left(t_{i+1}, \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}\right), \tag{39}$$

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

$$\underline{u}_1(t_i + \theta h_i) = d_{10}(\theta)\underline{y}_{1,i} + d_{11}(\theta)\underline{y}_{1,i+1} + h_i\left(d_{12}(\theta)\tilde{\underline{f}}_{1,i} + d_{13}(\theta)\tilde{\underline{f}}_{1,i+1}\right), \tag{40}$$

where, $d_{10}(\theta)$, $d_{11}(\theta)$, $d_{12}(\theta)$, and $d_{13}(\theta)$, are the standard Hermite basis polynomials obtained by requiring that $\underline{u}_1(t_i) = \underline{y}_{1,i}$, $\underline{u}_1(t_{i+1}) = \underline{y}_{1,i+1}$, $\underline{u}'_1(t_i) = \tilde{\underline{f}}_{1,i}$, and $\underline{u}'_1(t_{i+1}) = \tilde{\underline{f}}_{1,i+1}$. From standard theory for this type of interpolant it follows that (40) has an interpolation error that is $O(h_i^4)$.

Similarly, we can compute the endpoint stages,

$$\tilde{f}_{2,i} \equiv f_2\left(t_i, \underline{y}_{1,i}, \underline{y}_{2,i}, \underline{y}'_{2,i}\right), \quad \tilde{f}_{2,i+1} \equiv f_2\left(t_{i+1}, \underline{y}_{1,i+1}, \underline{y}_{2,i+1}, \underline{y}'_{2,i+1}\right), \tag{41}$$

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

$$\underline{u}_2(t_i + \theta h_i) = d_{20}(\theta)\underline{y}_{2,i} + d_{21}(\theta)\underline{y}_{2,i+1} +$$

$$h_i\left(d_{22}(\theta)\underline{y}'_{2,i} + d_{23}(\theta)\underline{y}'_{2,i+1}\right) + h_i^2\left(d_{24}(\theta)\tilde{\underline{f}}_{2,i} + d_{25}(\theta)\tilde{\underline{f}}_{2,i+1}\right), \tag{42}$$

where, $d_{20}(\theta), \ldots, d_{25}(\theta)$, are the standard Hermite basis polynomials obtained by requiring that $\underline{u}_2(t_i) = \underline{y}_{2,i}$, $\underline{u}_2(t_{i+1}) = \underline{y}_{2,i+1}$, $\underline{u}'_2(t_i) = \underline{y}'_{2,i}$, $\underline{u}'_2(t_{i+1}) = \underline{y}'_{2,i+1}$, $\underline{u}''_2(t_i) = \tilde{\underline{f}}_{2,i}$, and $\underline{u}''_2(t_{i+1}) = \tilde{\underline{f}}_{2,i+1}$. We can also form a Hermite interpolant, $\bar{\underline{u}}_2(t) \approx \underline{y}'_2(t)$, of the form,

$$\bar{\underline{u}}_2(t_i + \theta h_i) = \bar{d}_{20}(\theta)\underline{y}'_{2,i} + \bar{d}_{21}(\theta)\underline{y}'_{2,i+1} + h_i\left(\bar{d}_{22}(\theta)\tilde{\underline{f}}_{2,i} + \bar{d}_{23}(\theta)\tilde{\underline{f}}_{2,i+1}\right), \tag{43}$$

where, $\bar{d}_{20}(\theta)$, $\bar{d}_{21}(\theta)$, $\bar{d}_{22}(\theta)$, and $\bar{d}_{23}(\theta)$, are the standard Hermite basis polynomials that arise from requiring that $\bar{\underline{u}}_2(t_i) = \underline{y}'_{2,i}$, $\bar{\underline{u}}_2(t_{i+1}) = \underline{y}'_{2,i+1}$, $\bar{\underline{u}}'_2(t_i) = \tilde{\underline{f}}_{2,i}$, and $\bar{\underline{u}}'_2(t_{i+1}) = \tilde{\underline{f}}_{2,i+1}$. (We note that this implies that $d_{10}(\theta) \equiv \bar{d}_{20}(\theta)$, $d_{11}(\theta) \equiv \bar{d}_{21}(\theta)$, $d_{12}(\theta) \equiv \bar{d}_{22}(\theta)$, and $d_{13}(\theta) \equiv \bar{d}_{23}(\theta)$.) Standard theory for these types of interpolants states that (42) has an interpolation error of $O(h_i^6)$, and that (43) has an interpolation error that is $O(h_i^4)$.

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

When $k = 1$ or 2, the superconvergent collocation meshpoint approximations have errors that are $O(h_i^2)$ and $O(h_i^4)$, respectively, and therefore the Hermite interpolants, (40), (42), (43), can be used to provide SCIs. For $k = 3$, the superconvergent collocation meshpoint approximations have errors that are $O(h_i^6)$, and thus (42) can be used to provide an SCI for the $\underline{y}_2(t)$ approximation but (40) is not accurate enough to provide an SCI for $\underline{y}_1(t)$ nor is (43) accurate enough to provide one for $\underline{y}'_2(t)$. For $k = 4$, none of the Hermite interpolants are accurate enough to provide SCIs. (In these two latter cases, we will need to use the more general framework of CRK/CRKN methods, which we will discuss later in this section.)

3.3 Parameterized CRK and CRKN methods

For first order systems, (1), within the BVODE context, we have solution information associated with both endpoints of the ith subinterval, i.e., $\underline{y}_{1,i}$, $\underline{y}_{1,i+1}$. 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 [10]; with the additional restriction that the rth stage depend only on the previously computed stages, one gets the mono-implicit RK (MIRK) methods [8, 6]. By introducing weight polynomials, one can obtain continuous PIRK (CPIRK) methods; a subclass of these, the continuous MIRK methods, have been considered in [21].

The general form of a CPIRK method is the same as that of a CRK method, (33), and the definition of $\hat{t}_r$ is the same as in (26), but the $\underline{\hat{y}}_1, \ldots, \underline{\hat{y}}_s$ values are defined differently. Instead of the expression for $\underline{\hat{y}}_r$ given in (26), we have

$$\underline{\hat{y}}_r = (1 - v_r)\underline{y}_i + v_r\underline{y}_{i+1} + h_i \left(x_{r1}\underline{f}(\hat{t}_1, \underline{\hat{y}}_1) + \cdots + x_{rs}\underline{f}(\hat{t}_s, \underline{\hat{y}}_s) \right),$$

and the corresponding tableau for a CPIRK method is

$$\begin{array}{c|c|cccc} c_1 & v_1 & x_{11} & x_{12} & \ldots & x_{1s} \\ c_2 & v_2 & x_{21} & x_{22} & \ldots & x_{2s} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & v_s & x_{s1} & x_{s2} & \ldots & x_{ss} \\ \hline & & b_1(\theta) & b_2(\theta) & \ldots & b_s(\theta) \end{array} . \tag{44}$$

A CPIRK scheme can be converted to a standard CRK scheme; the A matrix appearing in (36) is given by $A = X + \underline{v}(\underline{b}(1))^T$, where A and X are s by s matrices whose r, jth elements are a_{rj} and x_{rj}, respectively, $\underline{v} = [v_1, \ldots, v_s]^T$ and $\underline{b}(1) = [b_1(1), \ldots, b_s(1)]^T$.

For second order BVODE systems, $\underline{y}''(t) = \underline{f}(t, \underline{y}(t), \underline{y}'(t))$, we have solution and some derivative information associated with the endpoints of the ith subinterval, i.e., $\underline{y}_{1,i}$, $\underline{y}_{2,i}$, $\underline{y}'_{2,i}$, $\underline{y}_{1,i+1}$, $\underline{y}_{2,i+1}$, $\underline{y}'_{2,i+1}$. 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 be computable in terms of previously computed stages, we get the mono-implicit RKN (MIRKN) methods; see, e.g., [20] and references within.

By introducing weight polynomials, one can obtain what are known as continuous PIRKN (CPIRKN) methods; a subclass of these, the continuous MIRKN methods, have been considered in [20]. The general form of a CPIRKN method is the same as that of a CRKN method, (34), and the definition of $\hat{t}_r$ is the same as in (26), 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 (31) and (30), we have

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

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

and

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

We represent the CPIRKN methods using a tableau of the form,

$$\begin{array}{c|c|c|cccc|c|cccc}
c_1 & v_1 & w_1 & x_{11} & x_{12} & \ldots & x_{1s} & v'_1 & x'_{11} & x'_{12} & \ldots & x'_{1s} \\
c_2 & v_2 & w_2 & x_{21} & x_{22} & \ldots & x_{2s} & v'_2 & x'_{21} & x'_{22} & \ldots & x'_{2s} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
c_s & v_s & w_s & x_{s1} & x_{s2} & \ldots & x_{ss} & v'_s & x'_{s1} & x'_{s2} & \ldots & x'_{ss} \\
\hline
 & & & b_1(\theta) & b_2(\theta) & \ldots & b_s(\theta) & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \ldots & \bar{b}_s(\theta)
\end{array}. \tag{45}$$

A CPIRKN scheme can be converted to a standard CRKN scheme; the A' matrix appearing in (37) is given by $A' = X' + \underline{v}'(\underline{\bar{b}}(1))^T$, where A' and X' are s by s matrices whose r,jth elements are a'_{rj} and x'_{rj}, respectively, $\underline{v}' = [v'_1, \ldots, v'_s]^T$ and $\underline{\bar{b}}(1) = [\bar{b}_1(1), \ldots, \bar{b}_s(1)]^T$, while the A matrix appearing in (37) is given by $A = X + \underline{v}(\underline{b}(1))^T + \underline{w}(\underline{\bar{b}}(1))^T$, where $\underline{w} = [w_1, \ldots, w_s]^T$.

It is possible to rewrite the Hermite interpolants, (40), (42), and (43), in terms of CPIRK/CPIRKN schemes, as follows. Because they are obtained from a collocation method, as explained in the previous section, the quantities,

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

satisfy (22), (23), and (24). 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 (40), (42), and (43). These equations can then be rewritten, respectively, as

$$\underline{u}_1(t_i+\theta h_i) = \underline{y}_{1,i} + h_i\left(d_{12}(\theta)\underline{\tilde{f}}_{1,i} + d_{13}(\theta)\underline{\tilde{f}}_{1,i+1} + \hat{d}_{11}(\theta)\hat{f}_{11} + \cdots + \hat{d}_{1k}(\theta)\hat{f}_{1k}\right), \tag{46}$$

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

$$\bar{\underline{u}}_2(t_i+\theta h_i) = \underline{y}'_{2,i} + h_i \left(\bar{d}_{22}(\theta)\tilde{\underline{f}}_{2,i} + \bar{d}_{23}(\theta)\tilde{\underline{f}}_{2,i+1} + \tilde{d}_{21}(\theta)\hat{f}_{21} + \cdots + \tilde{d}_{2k}(\theta)\hat{f}_{2k} \right), \tag{47}$$

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

$$\underline{u}_2(t_i + \theta h_i) = \underline{y}_{2,i} + \theta h_i \underline{y}'_{2,i} +$$

$$h_i^2 \left(d_{24}(\theta)\tilde{\underline{f}}_{2,i} + d_{25}(\theta)\tilde{\underline{f}}_{2,i+1} + \hat{d}_{21}(\theta)\hat{f}_{21} + \cdots + \hat{d}_{2k}(\theta)\hat{f}_{2k} \right), \tag{48}$$

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., (46), (47), (48), now fit into the standard forms for CPIRK and CPIRKN schemes discussed earlier in this subsection.

3.4 CPIRK/CPIRKN methods leading to SCIs

As mentioned earlier, Hermite interpolants cannot be used to derive methods that can form the basis for SCIs when $k \geq 3$. In this subsection, therefore, we will explain how to use CPIRK/CPIRKN methods in order to provide a framework for the development of SCIs for mixed first and second order systems when $k \geq 3$. As mentioned earlier, these methods will have embedded within them the collocation stages that were computed during the computation of the collocation solution.

On each subinterval, these CPIRK/CPIRKN methods will employ three kinds of stages:
(i) the first two stages will be the endpoint stages, (39), (41), which are easily computed from the discrete collocation solution and derivative values,
(ii) the next k stages will be the collocation stages, (9), which are already available from the collocation computation, and,
(iii) the rest of the stages will be MIRK/MIRKN stages; each such stage can depend on previously available stages from (i) and (ii), as well as previously computed MIRK/MIRKN stages.

These new MIRK/MIRKN stages will have the form,

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

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

and

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

where the first two stages, $\underline{f}_{lr}, l, r = 1, 2$, are the endpoint stages, (39), (41), the next k stages, $\underline{f}_{lr}, l = 1, 2, r = 3, \ldots, k+2$, are the collocation stages, (9), and the new MIRK/MIRKN stages are given by

$$\underline{f}_{lr} \equiv f_l\left(\hat{t}_r, \hat{\underline{y}}_{1r}, \hat{\underline{y}}_{2r}, \hat{\underline{y}}'_{2r}\right), \quad l = 1, 2, r = k+3, \ldots, s,$$

where $\hat{t}_r = t_i + c_r h_i$, and $\hat{\underline{y}}_{1r}$, $\hat{\underline{y}}_{2r}$, and $\hat{\underline{y}}'_{2r}$ are as defined above.

4 Derivation of CPIRK/CPIRKN methods leading to SCIs

In this section we derive specific CPIRK/CPIRKN methods, for $k = 1, 2, 3$ and 4, that can provide the basis for a corresponding set of SCIs for each k value.

4.1 Collocation with $k = 1$

When $k = 1$, the discrete collocation solutions and derivative meshpoint values have an error that is $O(h_i^2)$. Based on the theory discussed earlier in this report, the collocation polynomial approximating $\underline{y}_2(t)$ will have an error that is $O(h_i^p)$ where $p = \min(k+2, 2k)$, which equals 2 for this value of k. The collocation polynomials approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ have errors that are $O(h_i^p)$ where $p = \min(k+1, 2k)$, which again equals 2 for this value of k. Thus the collocation polynomials across the entire domain and the meshpoint values have the same order of accuracy.

We can improve upon the collocation polynomials only in terms of achieving higher continuity, which we can do by employing the Hermite interpolants (40), (42), (43).

Expressing these Hermite interpolants in the form of a CPIRKN method (as explained in the previous section), we get a CPIRKN method with the coefficient tableau:

$$\begin{array}{c|c|c|ccc|c|ccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{8} & 0 & 0 & 0 & \frac{1}{2} \\
\hline
 & & & b_1(\theta) & b_2(\theta) & b_3(\theta) & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta)
\end{array}, \tag{49}$$

where

$$\begin{array}{c|c|c}
r & b_r(\theta) & \bar{b}_r(\theta) \\
\hline
1 & -1/2\,\theta^2(\theta-1)^3 & \theta(\theta-1)^2 \\
2 & 1/2\,\theta^3(\theta-1)^2 & \theta^2(\theta-1) \\
3 & -1/2\,\theta^3(\theta-2) & -\theta^2(2\,\theta-3)
\end{array}. \tag{50}$$

This scheme, in addition to having the same order as the discrete collocation approximations at the meshpoints, 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 discrete collocation values have errors that are $O(h_i^4)$. The collocation polynomial approximating $\underline{y}_2(t)$ will have an error that is $O(h_i^p)$ where $p = \min(k+2, 2k)$, which equals 4 for this value of k. The collocation polynomials approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ have errors that are $O(h_i^p)$ where $p = \min(k+1, 2k)$, which equals 3 for this value of k. Thus the collocation polynomial approximating $\underline{y}_2(t)$ has the same order of accuracy across the entire domain as the discrete collocation values while the collocation polynomials approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ have errors that are one order of accuracy lower.

The use of (42) again allows us to improve the continuity of the $\underline{y}_2(t)$ approximation and we can use (40) and (43) to increase the order of accuracy and the continuity of the approximations to $\underline{y}_1(t)$ and $\underline{y}'_2(t)$.

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

$$\begin{array}{c|c|c|cccc|c}
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\
\frac{1}{2} - \frac{\sqrt{3}}{6} & 0 & 0 & 0 & 0 & \frac{1}{36} & \frac{5}{36} - \frac{\sqrt{3}}{12} & \dots \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & 0 & 0 & 0 & 0 & \frac{5}{36} + \frac{\sqrt{3}}{12} & \frac{1}{36} & \dots \\
\hline
 & & & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & \dots
\end{array} \tag{51}$$

$$\begin{array}{c|c|cccc}
\dots & 0 & 0 & 0 & 0 & 0 \\
\dots & 1 & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\
\dots & 0 & 0 & 0 & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\
\hline
\dots & & \bar{b}_1(\theta) & \bar{b}_2(\theta) & \bar{b}_3(\theta) & \bar{b}_4(\theta)
\end{array} , \tag{52}$$

where

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

. (53)

This scheme provides an approximation for $\underline{y}_2(t)$ that is C^2-continuous, and approximations for $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ that are C^1-continuous (whereas the corresponding collocation polynomials again have one order of continuity less in each case). As well, all approximations have the same order of accuracy across the entire domain as the discrete collocation approximations at the meshpoints.

4.3 Collocation with $k = 3$

In this case, the discrete collocation solution and derivative values have errors that are $\mathrm{O}(h_i^6)$, while the collocation polynomial approximating $\underline{y}_2(t)$ has an

error that is only $O(h_i^5)$ and the errors for the collocation polynomials approximating $\underline{y}_1(t)$ and $\underline{y}_2'(t)$ are only $O(h_i^4)$. While the use of (42) can provide an SCI of the appropriate order for $\underline{y}_2(t)$, the use of (40) and (43) to obtain approximations for $\underline{y}_1(t)$ and $\underline{y}_2'(t)$ will lead to interpolants that have errors that are only $O(h_i^4)$, and thus, in this case, the interpolants, (40) and (43), are not useful. We therefore now consider the direct derivation of a CPIRKN scheme which will provide SCIs for $\underline{y}_1(t)$ and $\underline{y}_2'(t)$, with C^1-continuity, whose errors are $O(h_i^6)$.

The derivation of a CPIRKN scheme is similar to the derivation of a CMIRKN scheme and we refer the reader to [20] for related detailed examples. We need this scheme to have a local error that is $O(h^6)$, consistent with the global error of the discrete solution; that is, using the terminology associated with Runge-Kutta methods, the CPIRKN must be of fifth order. (The definition of order for Runge-Kutta methods says that a pth order Runge-Kutta method has a local error that is $O(h^{p+1})$.)

We now discuss the derivation of this scheme. As indicated earlier in this report, the CPIRKN scheme will employ the two endpoint stages and the three available collocation stages. Considering just these first five stages, we observe that, for $q = 3$, the coefficients satisfy

$$X'\underline{c}^j + \frac{\underline{v}}{j+1} = \frac{\underline{c}^{j+1}}{j+1} \quad \text{for} \quad j = 0, 1, \ldots, q, \tag{54}$$

and

$$X\underline{c}^{j-1} + \frac{\underline{v}}{j+1} + \frac{\underline{\underline{w}}}{j} = \frac{\underline{c}^{j+1}}{j(j+1)} \quad \text{for} \quad j = 1, 2, \ldots, q, \tag{55}$$

where $\underline{c}^0$ 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, (54), (55), are called the *stage order conditions* up to order q - see [20].

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

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

and the single condition,

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

Since there are six independent conditions involving the $\bar{b}(\theta)$ polynomials, this implies that this CPIRKN scheme will require a total of six stages. Since the scheme will already use the two endpoint stages and the three collocation stages, one extra stage will be required. It will be a mono-implicit stage whose coefficients will satisfy the stage order conditions given above (so that no additional order conditions are introduced).

We will actually require the new sixth stage to have the maximum possible stage order, which turns out to be stage order six - that is, (54), (55), 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 $c_6 = \frac{1}{2} - \frac{\sqrt{10}}{10}$, $v_6 = c_6$, $w_6 = -c_6$, and ${v_6}' = c_6$. We then solve the above order conditions to obtain the $\underline{\bar{b}}(\theta)$ weight polynomials. As mentioned earlier, the $\underline{b}(\theta)$ weight polynomials for the approximation of $\underline{y}_2(t)$ can be determined using Hermite interpolation, and then the Hermite interpolant can be converted to CPIRKN form, using the approach described in the previous section.

The resultant scheme has the tableau:

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	$\ldots$
			$b_1(\theta)$	$b_2(\theta)$	$b_3(\theta)$	$b_4(\theta)$	$b_5(\theta)$	0	$\ldots$

$\ldots$	0	0	0	0	0	0	0	0
$\ldots$	1	0	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	0
$\ldots$	0	0	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$	0	0	, (56)
$\ldots$	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	
$\ldots$		$\bar{b}_1(\theta)$	$\bar{b}_2(\theta)$	$\bar{b}_3(\theta)$	$\bar{b}_4(\theta)$	$\bar{b}_5(\theta)$	$\bar{b}_6(\theta)$	

where $v_6 = \frac{1}{2} - \frac{\sqrt{10}}{10}$, $w_6 = -\frac{1}{2} + \frac{\sqrt{10}}{10}$, the values for x_{6r} and ${x'}_{6r}$ for $r = 1, \ldots, 5$, 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	$-\frac{29}{4800}\sqrt{2}\sqrt{3} + \frac{1709}{14400} - 1/36\sqrt{2}\sqrt{5}$	$\frac{1}{150}\sqrt{2}\sqrt{5} + \frac{3}{160}\sqrt{5}\sqrt{3}$, (57)
4	$\frac{407}{2250} - \frac{2}{45}\sqrt{10}$	$-\frac{19}{750}\sqrt{10}$
5	$\frac{29}{4800}\sqrt{2}\sqrt{3} + \frac{1709}{14400} - 1/36\sqrt{2}\sqrt{5}$	$\frac{1}{150}\sqrt{2}\sqrt{5} - \frac{3}{160}\sqrt{5}\sqrt{3}$

$b_6(\theta) \equiv 0$, the expressions for $b_r(\theta)$, for $r = 1, \ldots, 5$, are given by,

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

, (58)

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

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

. (59)

4.4 Collocation with $k = 4$

When $k = 4$, the discrete mesh point collocation solution and derivative values have errors that are $\mathrm{O}(h_i^8)$. On the other hand, the collocation polynomial approximating $\underline{y}_2(t)$ has an error that is only $\mathrm{O}(h_i^6)$ and the errors for the collocation polynomials approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$ are only $\mathrm{O}(h_i^5)$. In this subsection, we will derive a CPIRKN scheme that will provide an SCI approximating $\underline{y}_2(t)$, with C^2-continuity, whose error is $O(h_i^8)$, and SCIs approximating $\underline{y}_1(t)$ and $\underline{y}'_2(t)$, with C^1-continuity, whose errors are $O(h_i^8)$.

Since we want this CPIRKN scheme to have a local error that is $O(h_i^8)$, the CPIRKN must be of seventh order. This CPIRKN scheme will employ the two endpoint stages and the four collocation stages associated with the collocation computation. These first six stages have coefficients that satisfy the stage order conditions, (54) and (55), for $q = 4$. The additional MIRK/MIRKN stages will also be required to satisfy these stage order conditions. In fact, it will be useful to impose the stage order conditions with $q = 6$ on the MIRK/MIRKN stages.

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

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

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

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

$$\bar{\underline{b}}(\theta)\left(X'\left(X'\underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5}\right)\right) = 0,$$

$$\bar{\underline{b}}(\theta)\left(X'\underline{c}^5 + \frac{\underline{v}'}{6} - \frac{\underline{c}^6}{6}\right) = 0,$$

$$\bar{\underline{b}}(\theta)\left(\underline{c}\left(X'\underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5}\right)\right) = 0,$$

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

$$\underline{b}(\theta)\left(X'\underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5}\right) = 0.$$

The above conditions include a total of ten order conditions involving the $\bar{\underline{b}}(\theta)$ polynomials and a total of seven order conditions involving the $\underline{b}(\theta)$ polynomials. It would appear then that this CPIRKN scheme will require a total of ten stages, corresponding to the ten order conditions associated with the $\bar{\underline{b}}(\theta)$ polynomials. However, it turns out that we can satisfy the order condition,

$$\bar{\underline{b}}(\theta)\left(X'\left(X'\underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5}\right)\right) = 0, \tag{60}$$

through a careful choice of the available free coefficients. This allows us to reduce the number of order conditions from ten to nine which will then imply that the scheme will require only nine stages.

The details regarding how to choose the free coefficients to satisfy (60) are as follows. Letting

$$\underline{C}51 = X'\underline{c}^4 + \frac{\underline{v}'}{5} - \frac{\underline{c}^5}{5},$$

we observe that (60) can be rewritten as,

$$\bar{\underline{b}}(\theta)\,(X'\underline{C}51) = 0.$$

An examination of the vector $\underline{C}51$ shows that, due to the fact that the first two stages and all the MIRK/MIRKN stages satisfy the stage order conditions up to $q = 6$, all components of $\underline{C}51$ are zero except for the 3rd through 6th components (corresponding to the collocation stages). The next step is to then chose the available free coefficients that appear in X' and in $\underline{C}51$ in order to force $X'\underline{C}51 = \underline{0}$. The solution of the conditions $X'\underline{C}51 = \underline{0}$ together with the imposition of the requirement that the additional MIRK/MIRKN stages satisfy the stage order conditions (54) up to $q = 6$, forces $c_7 = \frac{1}{2} \pm \frac{\sqrt{7}}{14}$, with v'_7 left free; we choose $c_7 = \frac{1}{2} + \frac{\sqrt{7}}{14}$ and $v'_7 = c_7$. Also left free are c_8, v'_8, c_9, v'_9, and x'_{98}; we choose $c_8 = \frac{1}{5}$, $v'_8 = c_8$, $c_9 = \frac{4}{5}$, $v'_9 = c_9$, and $x'_{98} = 0$. With these choices of coefficients, $X'\underline{C}51 = \underline{0}$, and then the order condition, $\bar{\underline{b}}(\theta)\,(X'\underline{C}51) = 0$, is satisfied.

As mentioned above, this leaves nine order conditions involving the $\bar{b}(\theta)$ polynomials. Since this scheme will include the two endpoint stages and the four collocation stages, this means that three extra MIRK/MIRKN stages will be required. Applying the stage order conditions, (54), (55), leads to the specification of most of the remaining free coefficients but there are several free coefficients that remain; we 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 $\bar{\underline{b}}(\theta)$ polynomials and the seven order conditions involving the $\underline{b}(\theta)$ polynomials in order to obtain the $\bar{\underline{b}}(\theta)$ 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:

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

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

(61)

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 $c_7 = v_7 = \frac{1}{2} + 1/14\sqrt{7}$.

Since the expressions for the components of X and X' matrices and the coefficients of the $\underline{b}(\theta)$ and $\bar{\underline{b}}(\theta)$ polynomials are too complicated to present in exact form, we provide approximations to these values in terms of 20 digit decimal numbers.

The nine rows of the matrix X are

$$[0, 0, 0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, .32305531606773849038 \times 10^{-2}, -.12501978957195100 11 \times 10^{-2},$$

$$.599119902841667647 \times 10^{-3}, -.16908467308653538 \times 10^{-3}, 0, 0, 0],$$

$$[0, 0, .44654739516219931749 \times 10^{-1}, .11055161125036900810 \times 10^{-1},$$

$$-.15763439131757701 42 \times 10^{-2}, .31957112533586327 71 \times 10^{-3}, 0, 0, 0],$$

$$[0, 0, .10477319401975273714, .10928215124631229915,$$

$$.11055161125036900810 \times 10^{-1}, -.66685674525687900 5 \times 10^{-3}, 0, 0, 0],$$

$$[0, 0, .14960613448280714923, .19642483580315200379,$$

$$.83717022845102756843 \times 10^{-1}, .32305531606773849038 \times 10^{-2}, 0, 0, 0],$$

$$[.56407299162219992243 \times 10^{-2}, -.56407299162219991643 \times 10^{-2},$$

$$-.12818262090389426184 \times 10^{-1}, -.27667766089641676157 \times 10^{-1},$$

$$-.66656828962826040579 \times 10^{-1}, 0, 0, 0, 0],$$

$$[.46987308407158757980 \times 10^{-2}, -.14560641740492091960 \times 10^{-2},$$

$$-.18777651201651205208 \times 10^{-1}, -.41033995097025767855 \times 10^{-1},$$

$$-.23431020367989693539 \times 10^{-1}, 0, 0, 0, 0],$$

$$[.99888010024382092459 \times 10^{-2}, -.67461343357715427659 \times 10^{-2},$$

$$-.18777651201651205061 \times 10^{-1}, -.95324451125056381245 \times 10^{-2},$$

$$-.54932570352509823295 \times 10^{-1}, 0, 0, 0, 0].$$

The nine rows of the matrix X' are

$$[0, 0, 0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, .86963711284363464343 \times 10^{-1}, -.26604180084998793304 \times 10^{-1},$$
$$.12627462689404724524 \times 10^{-1}, -.355514968579568315 \times 10^{-2}, 0, 0, 0],$$

$$[0, 0, .18811811749986807165, .16303628871563653566,$$
$$-.27880428602470895218 \times 10^{-1}, .67355005945381555512 \times 10^{-2},$$
$$0, 0, 0],$$

$$[0, 0, .16719192197418877317, .35395300603374396654,$$
$$.16303628871563653566, -.14190694931141142966 \times 10^{-1}, 0, 0, 0],$$

$$[0, 0, .17748257225452261183, .31344511474186834680,$$
$$.35267675751627186462, .86963711284363464343 \times 10^{-1}, 0, 0, 0],$$

$$[.12595035543861662683 \times 10^{-1}, .27901157992841254519 \times 10^{-1},$$
$$.31424458236000028921 \times 10^{-1}, .12784556454961043482,$$
$$-.24867668578255783089 \times 10^{-1}, -.17489854774405759784, 0, 0, 0],$$

$$[-.41944590273292959284 \times 10^{-2}, -.16060145828848513267 \times 10^{-2},$$
$$.14012820352866098544, -.26472409697747613300 \times 10^{-1},$$
$$-.15474899161669444823, -.33179698061006715161 \times 10^{-1},$$
$$.80073369457001938508 \times 10^{-1}, 0, 0],$$

$$[.73255409726707038846 \times 10^{-2}, .99139854171151489343 \times 10^{-2},$$
$$.21610386356930788521 \times 10^{-1}, .80525407473982583723 \times 10^{-1},$$
$$-.47751174444964251771 \times 10^{-1}, -.15169751523273691284,$$
$$.80073369457001939545 \times 10^{-1}, 0, 0].$$

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

$$-1.4838054098817121549\,\theta^7 + 6.3599856012526592100\,\theta^6 - 10.677124344467704701\,\theta^5 +$$
$$8.7095135247042803904\,\theta^4 - 3.4085693716075227451\,\theta^3 + 0.50000000000000000000\,\theta^2,$$
$$2.0717501456738433973\,\theta^7 - 6.0844588431917852229\,\theta^6 + 6.3228756555322952849\,\theta^5 -$$
$$2.6793753641846084896\,\theta^4 + 0.36920840617025503042\,\theta^3,$$
$$2.4008645956962080391\,\theta^7 - 10.051375502307884471\,\theta^6 + 16.198264590214086232\,\theta^5 -$$
$$12.157487596307555198\,\theta^4 + 3.7715852335674557048\,\theta^3,$$

$$-1.5977758278295572808\,\theta^7+6.0738981481079401498\,\theta^6-8.3024639918680125568\,\theta^5+$$
$$4.4450128192755651476\,\theta^4-0.40020561139055488968\,\theta^3,$$
$$-6.0321156342090800130\,\theta^7+21.594087470436269686\,\theta^6-28.922144091532793184\,\theta^5+$$
$$17.193739762616692926\,\theta^4-3.7259604661751969129\,\theta^3,$$
$$-3.5901441705395393389\,\theta^7+10.917155179517231353\,\theta^6-11.659926172782139077\,\theta^5+$$
$$5.0666626066202185145\,\theta^4-0.72167134110935482879\,\theta^3,$$
$$8.2312263010898373512\,\theta^7-28.809292053814430705\,\theta^6+37.040518354904268003\,\theta^5-$$
$$20.578065752724593291\,\theta^4+4.1156131505449186412\,\theta^3,$$

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

$$118.21553917666580224\,\theta^7-450.21272045166368619\,\theta^6+677.90745020470247929\,\theta^5-$$
$$506.63265771593045627\,\theta^4+190.31615074107904624\,\theta^3-30.593761954853199618\,\theta^2+\theta,$$
$$-60.923872509999145945\,\theta^7+249.69188711833036825\,\theta^6-401.42828353803587618\,\theta^5+$$
$$316.73682438259717901\,\theta^4-122.33698407441238897\,\theta^3+18.260428621519868725\,\theta^2,$$
$$-312.89027249245509522\,\theta^7+1182.4848315645309359\,\theta^6-1760.0136798995063288\,\theta^5+$$
$$1290.0528958373880041\,\theta^4-465.90057061724230909\,\theta^3+66.440723029853561290\,\theta^2,$$
$$-305.59222210601679616\,\theta^7+1134.2402328807602768\,\theta^6-1652.2852242034705354\,\theta^5+$$
$$1181.2977566097015877\,\theta^4-415.39180675681081626\,\theta^3+58.057336153267600487\,\theta^2,$$
$$61.037789032562035884\,\theta^7-278.29971712366852415\,\theta^6+497.01008236447030656\,\theta^5-$$
$$432.96119140493011893\,\theta^4+182.24991389345060854\,\theta^3-28.710804184453033999\,\theta^2,$$
$$182.44470556590992181\,\theta^7-725.92534732162277999\,\theta^6+1143.7888217385069610\,\theta^5-$$
$$890.88946104215982562\,\theta^4+341.54246348060257557\,\theta^3-50.787254998668143015\,\theta^2,$$
$$214.91228070175434723\,\theta^7-752.19298245614029494\,\theta^6+1032.4385964912278336\,\theta^5-$$
$$700.61403508771909033\,\theta^4+238.12280701754382498\,\theta^3-32.666666666666657629\,\theta^2,$$
$$429.13801384671252243\,\theta^7-1603.2561966116421409\,\theta^6+2349.7734629640512849\,\theta^5-$$
$$1685.4482584736159590\,\theta^4+588.91084526808341633\,\theta^3-79.117866993589184628\,\theta^2,$$
$$-326.34196121513359226\,\theta^7+1243.4700124011158454\,\theta^6-1887.1912261219461249\,\theta^5+$$
$$1428.4581268946686793\,\theta^4-537.51281895229395736\,\theta^3+79.117866993589188390\,\theta^2.$$

5 Overview of the SCI Software

In this subsection we briefly describe the software which implements the CPIRKN methods presented in the previous section in order to obtain the corresponding SCIs. There are two subroutines: (i) an SCI setup routine, and (ii) an SCI evaluation routine.

(i) The SCI setup routine, SCISETUP(ISPACE,FSPACE), assumes that COLSYS/COLNEW has already been called to compute an approximate solution. SCISETUP is called once before any evaluations of the SCI are required. This routine employs a routine adapted from one in the MIRKDC package [11], and is called to set up the Runge-Kutta stages required by the CPIRKN scheme. The ISPACE and FSPACE arrays which this routine takes as input come from COLSYS/COLNEW. Extra information computed by the SCISETUP routine is stored at the end of the FSPACE array, for future use by the SCI evaluation routine.

The principle steps implemented in the SCISETUP setup routine are as follows:

1. Extract from the FSPACE vector, the discrete meshpoint solution and derivative information for each point of the mesh, i.e., $\underline{y}_{1,i}$, $\underline{y}_{2,i}$, $\underline{y}'_{2,i}$, $i = 0, \ldots, N$.

2. For each meshpoint, call the user-defined subroutine which evaluates the right-hand side of the ODE system, to get stage values at the mesh points corresponding to the discrete collocation solution and derivative values.

3. Extract from the FSPACE vector, the collocation stage information for each subinterval, i.e., (9), $i = 0, \ldots, N-1$.

4. For each subinterval, compute extra MIRK/MIRKN stages as needed using the solution and derivative endpoint information and endpoint and collocation stage information obtained in the previous steps, as well as previously computed MIRK/MIRKN stages.

(ii) The SCI evaluation routine, SCISLN(T,Z,ISPACE,FSPACE), assumes that an initial call to the SCISETUP routine has already taken place. The SCI evaluation routine is used instead of the APPSLN routine of COLSYS/COLNEW and is called once each time an evaluation of an SCI approximation to $\underline{z}(t)$ is required. This routine takes as input the evaluation point, T, and the ISPACE and FSPACE arrays, and returns the SCI approximation to the $\underline{z}(t)$ vector in the array Z.

6 Numerical Results

6.1 Test Problems

We have chosen three test problems from the literature: (a) a uniformly loaded beam problem from [3], (b) the swirling flow I problem from [4], and (c) the

swirling flow III problem from [4].

(a) The uniformly loaded beam problem consists of a single linear fourth order equation,

$$x^3u''''(x)+6x^2u'''(x)+6xu''(x)=1, \quad \text{with} \quad u(1)=u''(1)=u(2)=u''(2)=0, \tag{62}$$

which has an exact solution given by

$$u(x)=\frac{1}{4}(10\ln(2)-3)(1-x)+\frac{1}{2}\left(\frac{1}{x}+(3+x)\ln(x)-x\right). \tag{63}$$

We will rewrite this problem as two first order ODEs and one second order ODE as follows, letting $z_1(x)=u(x), z_2(x)=u'(x), z_3(x)=u''(x)$:

$$z_1{}'(x)=z_2(x), \quad z_2{}'(x)=z_3(x), \quad z_3{}''(x)=(1-6x^2z_4(x)-6xz_3(x))/x^3,$$

with the boundary conditions,

$$z_1(1)=z_3(1)=z_1(2)=z_3(2)=0.$$

(b) The swirling flow I problem consists of two nonlinear equations, one of third order, the other of second order; it has the form,

$$f'''(x)=\gamma^2-2f''(x)f(x)+(f'(x))^2-g^2(x), \quad g''(x)=2g(x)f'(x)-2f(x)g'(x), \tag{64}$$

with boundary conditions

$$f'(0)=0, \quad g(0)=1, \quad f(0)=0, \quad g(10)=\gamma, \quad f'(10)=0. \tag{65}$$

We choose $\gamma = 3.0$. This system has no known closed form exact solution; in order to estimate the errors in our approximate solutions, we computed a high-precision numerical solution to this problem using COLSYS/COLNEW. We will rewrite this problem as one first order ODE and two second order ODEs as follows, letting $z_1(x) = f(x), z_2(x) = f'(x), z_3(x) = f''(x), z_4(x) = g(x), z_5(x) = g'(x)$:

$$z_1{}'(x)=z_2(x), \quad z_2{}''(x)=\gamma^2-2z_3(x)z_1(x)+z_2(x)^2-z_4(x)^2,$$

$$z_4{}''(x)=2z_4(x)z_2(x)-2z_5(x)z_1(x),$$

with the boundary conditions,

$$z_2(0)=0, z_4(0)=1, z_1(0)=0, z_4(10)=\gamma, z_2(10)=0.$$

(c) The swirling flow III problem consists of one fourth order ODE and one second order ODE,

$$\epsilon f''''(x)=-f(x)f'''(x)-g(x)g'(x), \qquad \epsilon g''(x)=f'(x)g(x)-f(x)g'(x), \tag{66}$$

with boundary conditions

$$f(0) = f'(0) = f(1) = f'(1) = 0, \qquad g(0) = -1, \quad g(1) = 1. \tag{67}$$

We choose $\epsilon = 0.075$. As is the case for problem (b), since this system has no known closed form exact solution, we computed a high-precision numerical solution using COLSYS/COLNEW. We will convert this system to one consisting of two first order ODEs and two second order ODEs, letting $z_1(x) = f(x), z_2(x) = f'(x), z_3(x) = f''(x), z_4(x) = f'''(x), z_5(x) = g(x), z_6(x) = g'(x)$:

$$z_1{}'(x) = z_2(x), \quad z_2{}'(x) = z_3(x), \quad z_3{}''(x) = -z_1(x)z_4(x) - z_5(x)z_6(x),$$

$$z_5{}''(x) = z_2(x)z_5(x) - z_1(x)z_6(x),$$

with the boundary conditions,

$$z_1(0) = z_2(0) = z_1(1) = z_2(1) = 0, z_5(0) = -1, z_5(1) = 1.$$

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

6.2 Convergence Rates and Maximum Errors on Fixed Uniform Meshes

In order to numerically investigate the orders of convergence for the discrete collocation solutions, the collocation polynomials, and the superconvergent interpolants, we need to obtain solutions from COLSYS/COLNEW on fixed uniform meshes. To do this, COLSYS/COLNEW was run with communication parameters IPAR(8)=2 and IPAR(10)=2. This forces COLSYS/COLNEW to compute a solution on the given initial mesh, generate a new mesh by splitting each subinterval of the initial mesh in half, compute a second solution on the new mesh, generate an error estimate by comparing the two solutions, and then terminate. This allows us to control the meshes used by COLSYS/COLNEW so that comparisons of solutions obtained on uniform meshes of N and $2N$ subintervals, for experimental analysis of convergence rates, is possible. We consider meshes for which, N, the number of subintervals, is equal to 2, 4, 8, 16, 32, 64, and 128.

Recall that m_i is the order of the ith differential equation of the ODE system. Since COLSYS/COLNEW requires $k \geq \max_i m_i$, when we consider a system of BVODEs that includes at least one second order ODE (which implies that there exists an i such that $m_i = 2$), it follows that we must choose $k \geq 2$.

In this subsection we consider numerical results for the cases where $k = 3$ and 4, which are the ones where CPIRK/CPIRKN schemes must be employed to obtain an SCI. We compute collocation solutions using COLSYS/COLNEW

and then augment these with SCIs, using the algorithm described in the previous section.

We consider numerical experiments in which we compute the maximum error *over all solution components* for a given evaluation point. The Discrete Solution error is the maximum error of the collocation solution over all meshpoints. The errors for the Collocation Polynomial and for the Superconvergent Interpolant are the maximum errors of these continuous approximations over 10000 uniformly distributed sample points.

Tables 1, 2, and 3 give results for Problems (a), (b), and (c) for the $k = 3$ case. For this case, the theory indicates that we should expect error ratios that approach $2^6 = 64$ for the discrete collocation and SCI approximations (since the corresponding errors for these approximations are $O(h^6)$) and an error ratio that approaches of $2^4 = 16$ for the collocation polynomial approximations (since the largest of the errors for these approximations is $O(h^4)$). Tables 4, 5, and 6 give results for Problems (a), (b), and (c) for the $k = 4$ case. For this case, the error ratios for the discrete collocation and SCI approximations should approach $2^8 = 256$ (since the corresponding errors for these approximations are $O(h^8)$) while the error ratios for the collocation polynomial approximations should approach $2^5 = 32$ (since the largest of the errors for these approximations is $O(h^5)$) .

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(2.1\text{x}10^{-4})$	- $(8.5\text{x}10^{-3})$	- $(1.6\text{x}10^{-3})$
4	$45.7(4.6\text{x}10^{-6})$	$7.3(1.2\text{x}10^{-3})$	$26.1(6.2\text{x}10^{-5})$
8	$47.2(9.8\text{x}10^{-8})$	$10.2(1.1\text{x}10^{-4})$	$37.8(1.6\text{x}10^{-6})$
16	$63.1(1.5\text{x}10^{-9})$	$12.5(9.2\text{x}10^{-6})$	$48.0(3.4\text{x}10^{-8})$
32	$62.9(2.5\text{x}10^{-11})$	$14.1(6.5\text{x}10^{-7})$	$55.0(6.2\text{x}10^{-10})$
64	$63.9(3.8\text{x}10^{-13})$	$15.0(4.4\text{x}10^{-8})$	$59.2(1.0\text{x}10^{-11})$
128	$61.6(6.2\text{x}10^{-15})$	$15.5(2.8\text{x}10^{-9})$	$61.4(1.7\text{x}10^{-13})$
Theoretical	64	16	64

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

The main point of presenting these order of convergence results is to demonstrate that the derived schemes are achieving experimentally observable rates of convergence that are reasonably consist with those predicted by the theory. (As the errors approach values around 10^{-13} we start to see some degradation in the results due to the interference of round-off error.)

We also note that the error of the SCIs is typically two to three orders of magnitude smaller than that of the collocation polynomials. As well, we see that the error of the SCIs is typically an order of magnitude larger than that of the discrete collocation solution. Regarding this last observation, there are several ways to improve the accuracy of the SCIs. The free coefficients that arise during the derivation of the underlying CPIRKN schemes can be determined to

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(4.0\text{x}10^{-1})$	- $(7.9\text{x}10^{-1})$	- $(3.6\text{x}10^{0})$
4	$2.0(2.0\text{x}10^{-1})$	$2.7(3.0\text{x}10^{-1})$	$7.4(4.8\text{x}10^{-1})$
8	$7.9(2.5\text{x}10^{-2})$	$7.4(4.0\text{x}10^{-2})$	$15.1(3.2\text{x}10^{-2})$
16	$52.0(4.8\text{x}10^{-4})$	$13.1(3.1\text{x}10^{-3})$	$51.8(6.2\text{x}10^{-4})$
32	$94.4(5.1\text{x}10^{-6})$	$12.0(2.6\text{x}10^{-4})$	$63.8(9.7\text{x}10^{-6})$
64	$59.0(8.6\text{x}10^{-8})$	$12.8(2.0\text{x}10^{-5})$	$62.5(1.6\text{x}10^{-7})$
128	$64.8(1.3\text{x}10^{-9})$	$14.0(1.4\text{x}10^{-6})$	$64.1(2.4\text{x}10^{-9})$
256	$64.1(2.1\text{x}10^{-11})$	$15.1(9.5\text{x}10^{-8})$	$64.4(3.8\text{x}10^{-11})$
Theoretical	64	16	64

Table 2: Error ratios (maximum errors), $k = 3$, Problem (b)

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(3.3\text{x}10^{-2})$	- $(3.6\text{x}10^{-2})$	- $(8.4\text{x}10^{-2})$
4	$68.1(4.8\text{x}10^{-4})$	$4.2(8.5\text{x}10^{-3})$	$19.0(4.4\text{x}10^{-3})$
8	$39.2(1.2\text{x}10^{-5})$	$8.5(1.0\text{x}10^{-3})$	$35.9(1.2\text{x}10^{-4})$
16	$54.6(2.2\text{x}10^{-7})$	$11.4(8.8\text{x}10^{-5})$	$47.6(2.6\text{x}10^{-6})$
32	$61.3(3.6\text{x}10^{-9})$	$13.5(6.5\text{x}10^{-6})$	$55.1(4.7\text{x}10^{-8})$
64	$63.3(5.8\text{x}10^{-11})$	$14.7(4.4\text{x}10^{-7})$	$59.4(7.9\text{x}10^{-10})$
128	$64.0(9.0\text{x}10^{-13})$	$15.3(2.9\text{x}10^{-8})$	$61.4(1.3\text{x}10^{-11})$
Theoretical	64	16	64

Table 3: Error ratios (maximum errors), $k = 3$, Problem (c)

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(3.6\text{x}10^{-7})$	- $(1.4\text{x}10^{-3})$	- $(1.0\text{x}10^{-4})$
4	$171.6(2.1\text{x}10^{-9})$	$13.4(1.1\text{x}10^{-4})$	$74.3(1.4\text{x}10^{-6})$
8	$224.1(9.3\text{x}10^{-12})$	$19.4(5.5\text{x}10^{-6})$	$125.7(1.1\text{x}10^{-8})$
16	$245.9(3.8\text{x}10^{-14})$	$24.4(2.3\text{x}10^{-7})$	$174.2(6.2\text{x}10^{-11})$
32	$48.8(7.8\text{x}10^{-16})$	$27.8(8.2\text{x}10^{-9})$	$208.4(3.0\text{x}10^{-13})$
64	$0.8(1.0\text{x}10^{-15})$	$29.8(2.7\text{x}10^{-10})$	$4.2(7.0\text{x}10^{-14})$
128	$1.3(7.8\text{x}10^{-16})$	$30.9(8.9\text{x}10^{-12})$	$2.3(3.1\text{x}10^{-14})$
256	$1.4(5.6\text{x}10^{-16})$	$31.4(2.8\text{x}10^{-13})$	$1.6(1.9\text{x}10^{-14})$
Theoretical	256	32	256

Table 4: Error ratios (maximum errors), $k = 4$, Problem (a)

attempt to minimize the leading order terms in the error expansions for these schemes. As well, one could derive CPIRKN schemes of orders 6 and 8, rather

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(4.0\text{x}10^{-1})$	- $(4.4\text{x}10^{-1})$	- $(3.8\text{x}10^{0})$
4	$5.6(7.1\text{x}10^{-2})$	$5.4(8.2\text{x}10^{-2})$	$13.4(2.8\text{x}10^{-1})$
8	$90.0(7.9\text{x}10^{-4})$	$13.4(6.1\text{x}10^{-3})$	$50.5(5.6\text{x}10^{-3})$
16	$124.8(6.4\text{x}10^{-6})$	$15.2(4.0\text{x}10^{-4})$	$191.2(2.9\text{x}10^{-5})$
32	$383.1(1.7\text{x}10^{-8})$	$24.6(1.6\text{x}10^{-5})$	$295.5(9.9\text{x}10^{-8})$
64	$277.5(6.0\text{x}10^{-11})$	$30.2(5.4\text{x}10^{-7})$	$219.3(4.5\text{x}10^{-10})$
128	$246.4(2.4\text{x}10^{-13})$	$32.1(1.7\text{x}10^{-8})$	$260.4(1.7\text{x}10^{-12})$
256	$9.2(2.6\text{x}10^{-14})$	$31.9(5.3\text{x}10^{-10})$	$4.6(3.7\text{x}10^{-13})$
Theoretical	256	32	256

Table 5: Error ratios (maximum errors), $k = 4$, Problem (b)

N	Discrete Solution	Collocation Polynomial	Superconvergent Interpolant
2	- $(1.4\text{x}10^{-5})$	- $(1.5\text{x}10^{-3})$	- $(1.1\text{x}10^{-4})$
4	$1.0(1.4\text{x}10^{-5})$	$1.0(1.5\text{x}10^{-3})$	$1.0(1.1\text{x}10^{-4})$
8	$132.8(1.1\text{x}10^{-7})$	$18.6(8.1\text{x}10^{-5})$	$89.3(1.3\text{x}10^{-6})$
16	$212.1(5.0\text{x}10^{-10})$	$24.9(3.3\text{x}10^{-6})$	$150.6(8.3\text{x}10^{-9})$
32	$242.6(2.1\text{x}10^{-12})$	$28.6(1.1\text{x}10^{-7})$	$197.2(4.2\text{x}10^{-11})$
64	$29.9(6.9\text{x}10^{-14})$	$30.4(3.8\text{x}10^{-9})$	$57.3(7.4\text{x}10^{-13})$
128	$1.4(5.1\text{x}10^{-14})$	$31.2(1.2\text{x}10^{-10})$	$1.4(4.0\text{x}10^{-13})$
256	$1.2(4.4\text{x}10^{-14})$	$31.4(3.8\text{x}10^{-12})$	$2.1(1.9\text{x}10^{-13})$
Theoretical	256	32	256

Table 6: Error ratios (maximum errors), $k = 4$, Problem (c)

than the schemes presented here, which are of orders 5 and 7, in order to further reduce the error of the SCIs. We document these tasks, in the next section, as items for future work.

7 Summary, Conclusions, and Future Work

In this report we have described how to extend the work on the development of SCIs for Gaussian collocation solutions, reported in [12] from the first order BVODE system case, to the mixed first and second order BVODE system case. The approach involves augmenting the superconvergent collocation meshpoint values with interpolants that have the same superconvergent order.

For the low order cases ($k = 1, 2$), standard Hermite interpolants can be employed on each subinterval to obtain SCIs. For the $k = 3$ and 4 cases, a more general approach for the development of SCIs based on the use of CRKN methods is employed. Our numerical results show that it is possible to obtain

SCIs that are substantially more accurate than the corresponding continuous collocation solutions across the entire problem domain. These results also show that the SCIs have experimentally observable orders of accuracy that are resonably consistent with the expected orders, based on the theoretical framework that was used to derive them.

This report includes the development of CPIRKN methods that lead to SCIs of orders 2, 4, 6, and 8, and the approach considered in this report could be extended to derive higher order schemes. However, one could argue that it is sufficient to provide a range of SCIs of orders consistent with what is provided by COLSYS/COLNEW. For the COLSYS/COLNEW package, and for the mixed first and second order BVODE system case, the range of available k values is 2 to 7. For a given choice of k, the collocation solution at non-mesh points has an error that is $O(h^{k+2})$ for the solution components associated with the second order differential equations and an error that is $O(h^{k+1})$ for the solution components associated with the first order differential equations and the derivatives of the second order differential equations. This implies that the errors associated with the collocation solutions that can be obtained by applying COLSYS/COLNEW to a mixed first and second order BVODE system will be $O(h^4)$ through $O(h^9)$ for the solution components associated with the second order differential equations and $O(h^3)$ through $O(h^8)$ for the solution components associated with the first order differential equations. *Thus the range of orders of SCIs provided in this report is comparable to what is provided by COLSYS/COLNEW.* (Furthermore, there is little evidence to support the use of higher k values when one considers the accuracy attained compared to the work involved; for example, it is shown in [24], where Gaussian collocation is used for the spatial discretization of 1D PDEs, that even for very sharp tolerances the use of high order collocation methods is generally only justified up to methods having errors that are around $O(h^8)$.)

As mentioned earlier, additional work could be done in order to improve the accuracy of the CPIRKN methods upon which the SCIs are based. Future work could include further investigation of the SCIs developed here in order to optimize 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. Another possibility for future work in this direction could involve investigating the use of CPIRKN methods that are one order of accuracy higher than those presented in this report. A third possibility for attempting to improve the accuracy of the SCIs could involve looking at expressing the polynomial weights in a barycentric form rather than in terms of a monomial basis. The paper [16] shows that the use of a barycentric representation can lead to less interference from round-off error.

A major direction for future work is to investigate the extension of the work reported in [1] in order to incorporate the schemes described in this report to allow COLNEWSC to directly treat mixed first and second order BVODE systems. (In the current version of COLNEWSC, higher order systems are converted to first order so that the approach developed in [12] can be employed.)

Another possible direction for future work is to explore the idea discussed

in the Introduction which would involve modifying COLSYS/COLNEW so that error control is applied only to the discrete superconvergent collocation values at the mesh points. Since these values are much more accurate, this would allow the code to converge much more quickly to a numerical solution that satisfies the user tolerance, and then an SCI could be employed at the end of the computation to extend the discrete solution to a continuous solution of the same order of accuracy across the domain. As mentioned earlier however, this would mean that the error of the returned SCI would not be controlled.

Finally, as mentioned earlier, the collection of BVODEs given in [4] involves differential equations with solution derivatives in the range from 1 to 4, and COLSYS/COLNEW is able to treat mixed order systems across this range. This suggests extending the approach considered in this report to derive methods leading to SCIs for mixed order systems having derivatives over this range. In order to be able to obtain SCIs for BVODE systems that include solution derivatives of orders 3 and 4, the primary challenge will be to extend the classes of CRK and CRKN methods to develop methods for differential equations in which derivatives of orders 3 and 4 appear. With this goal in mind, we have developed extensions of the continuous MIRK (CMIRK) methods for the direct treatment of systems of differential equations involving derivatives of orders 3 and 4. See the Appendix for a description of these methods.

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.

[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.

[3] U.M. Ascher, J. Christiansen, R.D. Russell, Collocation software for boundary value ODEs, ACM Trans. Math. Softw., 7, 1981, 209–222.

[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, SIAM, Philadelphia, 1995.

[5] 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.

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

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

[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.

[9] 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.

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

[11] W.H. Enright, P.H. Muir, Runge-Kutta software with defect control for boundary value ODEs, SIAM J. Sci. Comput., 17, 1996, 479-497.

[12] W.H. Enright, P.H. Muir, Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations, SIAM J. Sci. Comput., 21, 1999, 227–254.

[13] W.H. Enright, R. Sivasothinathan, Superconvergent interpolants for collocation methods applied to mixed order BVODES, ACM Trans. on Math. Softw., 26, 2000, 323–351.

[14] J.M. Fine, Interpolants for Runge-Kutta-Nyström methods, Computing, 39, 1987, 27–42.

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

[16] N.J. Higham, The numerical stability of barycentric Lagrange interpolation, IMA J. Numer. Anal., 24, 2004, 547–556.

[17] H. Jin, S. Pruess, Uniformly superconvergent approximations for linear two-point boundary value problems, SIAM J. Numer. Anal., 35, 1998, 363–375.

[18] S. Karlin, J.M. Karon, On Hermite-Birkhoff interpolation, J. Approx. Theory, 6, 1972, 90–115.

[19] A. Marthinsen, Continuous extensions to Nyström methods for second order initial value problems, BIT, 36, 1996, 309–332.

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

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

[22] B. Owren, M. Zennaro, Order barriers for continuous explicit Runge-Kutta methods, Math. Comp., 56, 1991, 645–661.

[23] B. Owren, M. Zennaro, Derivation of optimal continuous explicit Runge-Kutta methods, SIAM J. Sci. Stat. Comp., 13, 1992, 1488–1501.

[24] 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.

[25] S. Pruess, Interpolation schemes for collocation solutions of two point boundary value problems, SIAM J. Sci. Stat. Comput., 7, 1986, 322-333.

[26] S. Pruess, H. Jin, A stable high-order interpolation scheme for superconvergent data, SIAM J. Sci. Comput., 17, 1996, 714–724.

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

[28] R. Weiss, The application of implicit Runge-Kutta and collocation methods to boundary value problems, Math. Comp., 28, 1974, 449-464.

8 Appendix

8.1 Generalized CMIRK schemes for third order systems

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

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

Such methods are extensions of the continuous MIRKN methods mentioned earlier in this report. It is straightforward to extend these methods to CPIRKN methods of the same type which could then be used to provide SCIs for this problem class.

The general form for these methods is,

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

where

$$k_r = f(\hat{t}_r, \hat{y}_r, \hat{y}_r', \hat{y}_r''), \quad \hat{t}_r = t_{i-1} + c_r h_i,$$

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

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

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

8.2 Generalized CMIRK schemes for fourth order systems

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 of these methods is,

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

where

$$k_r = f(\hat{t}_r, \hat{y}_r, \hat{y}'_r, \hat{y}''_r, \hat{y}'''_r), \quad \hat{t}_r = t_{i-1} + c_r h_i,$$

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

$$\begin{aligned}
\hat{y}'_r = {} & (1 - v'_r) y'_{i-1} + v'_r y_{i+1} + h((c_r - v'_r - w'_r) y''_i + w'_r y''_{i+1}) + \\
& h_i^2 \left(\left(\frac{c_r^2}{2} - \frac{v'_r}{2} - w'_r - u'_r \right) y'''_i + u'_r y'''_{i+1} \right) + h_i^3 \sum_{j=1}^{r-1} x'_{rj} k_j,
\end{aligned}$$

$$\begin{aligned}
\hat{y}'_r = {} & (1 - v''_r) y''_i + v''_r y'_{i+1} + \\
& h_i \left((c_r - v''_r - w''_r) y'''_i + w''_r y'''_{i+1} \right) + h_i^2 \sum_{j=1}^{r-1} x''_{rj} k_j,
\end{aligned}$$

$$\hat{y}'''_r = (1 - v'''_r) y'''_i + v'''_r y'''_{i+1} + h_i \sum_{j=1}^{r-1} x'''_{rj} k_j.$$