

New Interpolants for Asymptotically Correct Defect Control of BVODEs *

W.H. Enright[†] and P.H. Muir[‡]

Abstract

The defect of a continuous approximate solution to an ODE is the amount by which that approximation fails to satisfy the ODE. A number of studies have explored the use of *asymptotically correct* defect estimates in the numerical solution of initial value ODEs (IVODEs). By employing an appropriately constructed interpolant to an approximate discrete solution to the ODE, various researchers have shown that it is possible to obtain estimates of the local error and/or the maximum defect that are asymptotically correct on each step, as the stepsize $h \rightarrow 0$. In this paper, we investigate the usefulness of asymptotically correct defect estimates for defect control in boundary value ODE (BVODE) codes. In the BVODE context, for a sequence of meshes which partition the problem interval, one computes a discrete numerical solution, constructs an interpolant, and estimates the maximum defect. The estimates (typically obtained by sampling the defect at a small number of points on each subinterval of the mesh) are used in a redistribution process to determine the next mesh and thus the availability of these more reliable maximum defect estimates can lead to improved meshes. As well, when such estimates are available, the code can terminate with more confidence that the defect is bounded throughout the problem domain by the user-prescribed tolerance. In this paper we employ a boot-strapping approach to derive interpolants that allow asymptotically correct defect estimates. Numerical results are included to demonstrate the validity of this approach.

Subject Classification: 65L05, 65L10.

Keywords: Runge-Kutta schemes, boundary value ODEs, defect control.

1 Introduction

Systems of boundary value ordinary differential equations (BVODEs) arise in a wide variety of applications - see [1], section 1.2. We will assume BVODEs of

*This work was supported by the Natural Sciences and Engineering Research Council of Canada.

[†]Department of Computer Science, University of Toronto, Toronto Ontario, Canada, M5S 2E4, enright@cs.toronto.edu

[‡]Department of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, Canada, B3H 3C3, muir@smu.ca.

the form,

$$y'(t) = f(t, y(t)), \quad g(y(a), y(b)) = 0, \quad (1)$$

where $t \in [a, b]$, $y: \mathbb{R} \rightarrow \mathbb{R}^n$, $f: \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, and $g: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. We refer the reader to [1], section 1.1, for a description of various classes of BVODEs and a discussion of how many of these can be converted to an equivalent problem of the form given in (1). Necessary and sufficient conditions for the existence and uniqueness of solutions to (1) are discussed in [1], section 3.1.

Runge-Kutta (RK) schemes (see, e.g., [5]) have been used in the numerical solution of BVODEs for some time (see, e.g., [35]). The COLSYS/COLNEW codes [2], [3], the COLMOD code [8] and the NAG code, D02TKF [29], are based on collocation methods, which for (1) are equivalent to a subclass of RK methods [36]. Another popular subclass of RK methods used for BVODEs are the mono-implicit Runge-Kutta (MIRK) methods (see [6], [9], [10], [11], [7], [19], [23], [4]). Specific A-stable, symmetric methods from this subclass have been employed in the Fortran 77 BVODE packages TWPBVP [12] and MIRKDC [19], in the MATLAB solver bvp4c [26], and in the recently developed Fortran 90/95 solver BVP_SOLVER [34].

All of the above codes compute an approximate solution to the BVODE by employing a mesh of points that partitions the problem domain into subintervals. The codes iteratively adapt the mesh based on some measure of the accuracy of the numerical solution on each subinterval, with the goal of obtaining a numerical solution for which the estimated accuracy of the solution is less than a user-provided tolerance on each subinterval. Codes such as COLSYS, COLNEW, COLMOD, D02TKF and TWPBVP use some estimate of the *global error* of the numerical solution; that is, the magnitude of the difference between the true solution and the approximate solution as the underlying measure of accuracy. Other codes such as MIRKDC, bvp4c, and BVP_SOLVER use an estimate of the magnitude of the *defect* of the numerical solution as the underlying measure of accuracy. Letting $u(t)$ be a continuous approximate solution generated by the method, then, for the ODE (1), the defect is $\delta(t) = u'(t) - f(t, u(t))$, i.e., the amount by which the approximate solution fails to satisfy the ODE. For a method that controls an estimate of the defect, it is also useful to efficiently estimate the sensitivity or conditioning of the BVODE [33] since the product of the associated conditioning constant and the maximum defect can be shown to give a bound on the global error [1].

The goal in *defect control* is to compute a numerical solution by adaptively choosing a mesh so that the maximum defect over the entire problem domain is bounded by a user-provided tolerance. In order to do this, one needs an estimate of the maximum defect on each subinterval. While it is straightforward to compute $\delta(t)$ at any point in the domain, the challenge is to determine in an efficient manner, the maximum value of the defect on each subinterval. When a standard interpolant is employed for $u(t)$ the usual approach is to simply sample the defect at a few points on each subinterval with the hope that one of the points will be close enough to the location of the true maximum defect. When the number of samples is small (and this must be the case in order for the

estimation process to be reasonably efficient) there is no particular justification for this hope. Since such sampling will essentially always underestimate the maximum defect, it is easy to see that defect control codes can return solutions whose defects do not meet the user tolerance. When standard defect estimation is employed we have observed in some cases that the estimated maximum defect can be exceeded by the true maximum defect by an order of magnitude. Thus a defect control code may accept a numerical solution for which the defect is in fact larger than the user-tolerance. Furthermore, the underestimate of the maximum defect can impact negatively on the performance of the mesh selection algorithm because it will not have access to a good “profile” of the defect over the subintervals of the mesh.

The primary goal of this paper is to consider the development of new types of interpolants that can replace the standard ones currently employed in defect control BVODE solvers such as MIRKDC and BVP_SOLVER. The new interpolants will be shown to yield defects for which *the location of the maximum defect on each subinterval can be determined à priori* (at least asymptotically) allowing a code employing these interpolants to terminate with more confidence that the defect is within the prescribed tolerance. As well, the availability of more reliable estimates of the maximum defect on each mesh subinterval can improve the mesh redistribution process, thus contributing to an improvement in the overall efficiency of the computation.

In the approach we consider in this paper for the numerical solution of a BVODE, one begins with a MIRK formula that produces approximations at the discrete mesh points $a = t_0 < t_1 \cdots < t_N = b$. One then introduces a local interpolant, $u_i(t)$, associated with the i^{th} subinterval $[t_{i-1}, t_i]$, which will interpolate the discrete solution at the endpoints t_{i-1} and t_i and will determine an accurate approximation for any $t \in [t_{i-1}, t_i]$. The associated numerical solution, $u(t)$ is then the piecewise polynomial defined by the collection of interpolants, $\{u_i(t)\}_{i=1}^N$. For a given discrete MIRK formula there will be a large family of possible interpolation schemes to adopt and we will describe and justify our particular choice later in this paper. Our choice is primarily guided by our desire to have an accurate $u_i(t)$ for which it is possible to derive an inexpensive and asymptotically correct estimate of the maximum defect associated with the i^{th} subinterval.

One can analyze the error and defect associated with $u_i(t)$ by considering it to be an approximation to the solution, $z_i(t)$, of the local IVP,

$$z_i' = f(t, z_i), \quad z_i(t_{i-1}) = y_{i-1}, \quad t \in [t_{i-1}, t_i]. \quad (2)$$

For this analysis it is convenient to introduce the interpolant, $\tilde{z}_i(t)$, of degree at most $p + 1$, that interpolates the exact local solution at t_{i-1}, t_i , and the derivative of the exact local solution at the p distinct points $t_{i-1}, t_i, t_{i-1} + \mu_r h_i$, for $r = 1, 2, \dots, (p - 2)$. The free parameters, $0 < \mu_1 \cdots < \mu_{p-2} < 1$ will depend on the specific local interpolant being analyzed. This approach allows one to analyze the error in $u_i(t)$ by considering two components – the error inherent in polynomial interpolation (the local interpolation error) and the error that

arises as a consequence of “inexact” values being interpolated (the data error associated with the fact that we are interpolating approximate solution and derivative values). This approach was first used for the analysis of continuous Runge Kutta methods for IVPs by Shampine [31] and Gladwell et al. [22] to investigate the local errors introduced on each step of an IVP method. It was also subsequently used by Higham ([24], [25]) to investigate the defects and the quality of the defect estimates for a different class of continuous Runge Kutta methods for IVPs.

One of the first papers to consider the defect of the numerical solution in the BVODE context was [30]. In a series of papers, [13], [14], [15], Enright considers the use of defect control in the numerical solution of IVPs. As mentioned earlier, defect control for BVODEs is considered in Fortran 77 MIRKDC code [20] and in the MATLAB bvp4c code [26]; the latter employs an interpolant that allows an asymptotically correct estimate of the maximum defect but the interpolant is not of optimal order. The paper, [16], surveys work up to the year 2000 on the use of defect control for the IVPs and BVODEs and also includes some discussion of defect control for delay differential equations (DDEs) and differential algebraic equations (DAEs). The paper, [32], discusses a new MATLAB DDE solver that employs defect control. The recent paper, [18], considers asymptotically correct defect control on problems where the local interpolation error is assumed to be small relative to the data errors. In [21], the authors have derived and justified interpolation schemes allowing asymptotically correct defect estimates for optimal p^{th} -order continuous Runge Kutta methods for $4 \leq p \leq 8$ without the assumption on the local interpolation error.

In most of these investigations, the focus has been on explicit Runge Kutta methods; asymptotically correct expressions for the continuous local error, $z_i(t) - u_i(t)$, and for the defect, $u_i'(t) - f(t, u_i(t))$, are derived and the relationship between these quantities and the discrete local error investigated. The basic idea in the development of an interpolant that allows an asymptotically correct estimate of the defect depends on the relationship between the local error and the defect. The key tool is the well-known error expansion for Runge-Kutta methods - see, e.g., [5]. For the methods we consider, on the i^{th} subinterval, $u_i(t)$ satisfies

$$u_i(t) - z_i(t) = O(h_i^{p+1}),$$

where $z_i(t)$ satisfies (2) and $h_i = t_i - t_{i-1}$. Similarly, the derivative of the numerical solution satisfies

$$u_i'(t) - z_i'(t) = O(h_i^p).$$

Using the fact that $z_i(t)$ is the exact solution of (2), the defect of the numerical solution can be written, for $t \in [t_{i-1}, t_i]$, as

$$\delta_i(t) = (u_i'(t) - z_i'(t)) - (f(t, u_i(t)) - f(t, z_i(t))). \quad (3)$$

Assuming a Lipschitz condition on f , the second term in (3) is $O(h_i^{p+1})$ and we have

$$\delta_i(t) = (u_i'(t) - z_i'(t)) + O(h_i^{p+1}).$$

From this expression we can see that the expansion of the defect has a leading term that is $O(h_i^p)$ but in this paper we will examine a more detailed form of this expression. This relationship will be used to identify those interpolation schemes for which inexpensive asymptotically correct estimates of the maximum defect are available. In our justification of a suitable $u_i(t)$ for use in BVODE methods we will adopt this approach but we point out that the underlying discrete MIRK formula used for the determination of the mesh point solution approximations, $\{y_i\}_{i=0}^N$, will be implicit while the $u_i(t)$ will be determined afterwards from an explicit computation.

This paper is organized as follows. In the next section we briefly review MIRK and continuous MIRK (CMIRK) [28] schemes, as well as a *boot-strapping* approach for deriving higher order Hermite-Birkhoff interpolants. We also contrast the continuity of CMIRK and Hermite-Birkhoff interpolants in the BVODE context. This is followed in Section 3 by an analysis of the form of the defect for a standard CMIRK interpolant and for a special Hermite-Birkhoff interpolant we derive, which is shown to allow an asymptotically correct defect estimate. Section 4 provides numerical results which confirm the validity of our approach and Section 5 gives our summary and conclusions.

2 MIRK and CMIRK schemes

In the MIRKDC and BVP_SOLVER codes, the primary formulas used for the discretization of the BVODEs and for the construction of the continuous solution approximations are MIRK and CMIRK formulas, which we now briefly review.

A discrete numerical solution, $y_i \approx y(t_i)$, $i = 0, \dots, N$, is obtained by applying Newton's method to the nonlinear system of equations consisting of the boundary conditions and n equations per subinterval having the following form (on the i^{th} subinterval),

$$y_i - y_{i-1} - h_i \sum_{r=1}^s b_r k_r = 0, \quad (4)$$

where $h_i = t_{i-1} - t_i$ and

$$k_r = f \left(t_{i-1} + c_r h_i, (1 - v_r) y_{i-1} + v_r y_i + h_i \sum_{j=1}^{r-1} x_{rj} k_j \right). \quad (5)$$

The scheme is defined by the number of stages, s , the coefficients, $\{v_r\}_{r=1}^s$ and $\{x_{rj}\}_{j=1, r=1}^{r-1, s}$, and the weights $\{b_r\}_{r=1}^s$. The abscissa, $\{c_r\}_{r=1}^s$, are defined by $c_r = v_r + \sum_{j=1}^{r-1} x_{rj}$. The coefficients of a MIRK scheme are usually presented

in a (Butcher) tableau of the form,

c_1	v_1	0	0	0
c_2	v_2	x_{21}	0	0
\vdots	\vdots	\vdots	\ddots			\vdots
\vdots	\vdots	\vdots		\ddots		\vdots
c_s	v_s	x_{s1}	x_{s2}	...	$x_{s,s-1}$	0
		b_1	b_2	b_s

A MIRK scheme is of order p if the numerical solution at the i^{th} meshpoint, y_i , obtained by solving (4), satisfies

$$|z_i(t_i) - y_i| = O(h_i^{p+1}).$$

A family of MIRK schemes of order p is derived by requiring its coefficients to satisfy a set of equations called *order conditions* - see, e.g., [5].

After the discrete solution is obtained, a CMIRK scheme can be used on each subinterval to augment the discrete solution with a C^1 continuous interpolant over the whole problem interval. A CMIRK scheme applied on the subinterval $[t_{i-1}, t_i]$, gives,

$$u_i(t_{i-1} + \theta h_i) = y_{i-1} + h_i \sum_{r=1}^{s^*} b_r(\theta) k_r, \quad (6)$$

for $0 \leq \theta \leq 1$, where the k_r 's are defined as in (5) and s^* is the total number of required stages. In addition to the coefficients which define its stages, the scheme is defined by the weight polynomials, $\{b_r(\theta)\}_{r=1}^{s^*}$, which are polynomials in θ . Some computational savings are achieved by deriving CMIRK schemes for which have their first s stages identical to those of the MIRK scheme; the underlying discrete MIRK scheme is then said to be "embedded" within the CMIRK scheme. A CMIRK scheme is of order p if, with $u_i(t)$ as in (6), we have

$$\max_{0 \leq \theta \leq 1} |z_i(t_{i-1} + \theta h_i) - u_i(t_{i-1} + \theta h_i)| = O(h_i^{p+1}).$$

A p^{th} order CMIRK scheme can be derived by requiring the coefficients and weight polynomials to satisfy continuous versions of the MIRK order conditions (see [28]).

An alternative approach for the derivation of an interpolant for the discrete numerical solution associated with a Runge-Kutta method is based on an algorithm known as boot-strapping [17]. Such interpolants, $\tilde{u}_i(t)$, are usually derived in a Hermite-Birkhoff form: on the subinterval $[t_{i-1}, t_i]$, and for $0 \leq \theta \leq 1$, we have

$$\tilde{u}_i(t_{i-1} + \theta h_i) = d_0(\theta) y_{i-1} + d_1(\theta) y_i + h_i \sum_{r=1}^{\tilde{s}^*} \tilde{b}_r(\theta) k_r, \quad (7)$$

where the k_r 's have the same general form as in (5), the weight polynomials, $d_0(\theta), d_1(\theta), \{\tilde{b}_r(\theta)\}_{r=1}^{\tilde{s}^*}$, are polynomials in θ , and \tilde{s}^* is the total number of

required stages. It is straightforward to convert an interpolant represented in this form into the standard CMIRK representation (6). To see this, use the discrete formula (4) to substitute for y_i in (7), giving (7) in the CMIRK form

$$\tilde{u}(t_{i-1} + \theta h_i) = y_{i-1} + h_i \sum_{r=1}^{s^*} \left(b_r d_1(\theta) + \tilde{b}_r(\theta) \right) k_r, \quad (8)$$

where we have made use of the fact that $d_0(\theta) + d_1(\theta) = 1$ (which follows from the interpolation conditions imposed on (7)).

From the above it is easy to see that one could derive an interpolant in Hermite-Birkhoff form (7) and convert it to the CMIRK form (6) assumed by MIRKDC and BVP_SOLVER. However, there is some advantage to employing the Hermite-Birkhoff form (7) with a BVODE code. In the numerical solution of a BVODE, the solution approximations at the mesh points, $\{y_i\}_{i=0}^N$, are obtained from the solution of a large nonlinear algebraic system which consists of the boundary conditions together with $N - 1$ sets of equations of the form (4) and thus *given pairs of y_{i-1} and y_i values satisfy (4) only to within the Newton tolerance applied to this nonlinear system.* The CMIRK interpolant is constructed with $k_1 = f(t_{i-1}, y_{i-1})$ and $k_2 = f(t_i, y_i)$, and so that $u'(t_{i-1}) = f(t_{i-1}, y_{i-1})$ and $u'(t_i) = f(t_i, y_i)$ giving continuity of $u'(t)$ over $[a, b]$. The weight polynomials, $b_r(\theta)$, satisfy $b_r(0) = 0$ giving $u(t_{i-1}) = y_{i-1}$ but for $\theta = 1$ the CMIRK interpolant reduces to the discrete MIRK formula associated with (4), i.e., $u_i(t_i) = y_{i-1} + h \sum_{r=1}^s b_r k_r$ and as we point out above this implies only that $u(t_i) \approx y_i$ (with a potential disagreement of the magnitude of the Newton tolerance.) *Thus $u(t)$ may have discontinuities at the mesh points of the size of the Newton tolerance when the CMIRK representation of the interpolant is employed.*

On the other hand, when we use the Hermite-Birkhoff representation of the interpolant, we have k_1 and k_2 as above and $\tilde{u}'(t_{i-1}) = f(t_{i-1}, y_{i-1})$ and $\tilde{u}'(t_i) = f(t_i, y_i)$ so that we have continuity of $\tilde{u}'(t)$ over $[a, b]$ as before but we also have $\tilde{u}(t_{i-1}) = y_{i-1}$ and $\tilde{u}(t_i) = y_i$ and thus *continuity of $\tilde{u}(t)$ is also obtained.* In summary, the explicit appearance of y_i in the Hermite-Birkhoff form ensures $\tilde{u}(t) \in C^1[a, b]$ but since (4) will only be satisfied to within the Newton tolerance, we have $u(t) \notin C^1[a, b]$ (with discontinuities in $u(t)$ at the mesh points of the magnitude of the Newton tolerance.)

In [17] the authors developed the boot-strapping algorithm to derive a p^{th} order local interpolant for an arbitrary p^{th} order discrete Runge Kutta formula. Starting with a 4^{th} order local (Hermite) interpolant, $u_{4,i}(t)$ (easily constructed for any RK method) they show how to define the extra explicit stages necessary to define a sequence of more accurate local interpolants $u_{q,i}(t)$, $q = 5, 6 \dots p$, where $u_{q,i}(t)$ is an $O(h_i^{q+1})$ approximation to the local solution on the i^{th} interval. We will use a similar idea, starting with a standard p^{th} order CMIRK, such as that already used in MIRKDC or BVP_SOLVER, and use it to derive an “improved” local Hermite-Birkhoff interpolant that allows an inexpensive asymptotically correct estimate of the maximum magnitude of the defect.

3 Derivation of Interpolants Allowing Asymptotically Correct Defect Estimates

The BVP_SOLVER code provides the user with the option of employing second order, fourth order, or sixth order Runge-Kutta formulas. We will illustrate our approach for the derivation of an interpolant for which the maximum defect can be reliably estimated by considering in detail the derivation of such an interpolant for the sixth order case, as this is the most challenging of the three cases that need to be developed. However, for second and fourth order cases improved local interpolants can be developed similarly, as is shown for IVPs in [21].

The formulas that are implemented in BVP_SOLVER have been optimized with respect to their leading error coefficients. That is the free coefficients of the Runge-Kutta formulas that remained after the order conditions were satisfied were chosen to minimize the norm of the coefficients of the elementary differentials appearing in the $O(h_i^{p+1})$ term of the error expansion subject to the constraint that the ratio of this norm to the norm of the coefficients of the elementary differentials associated with the $O(h_i^{p+2})$ term be not too small. See [27] for details.

3.1 Analysis of the Standard Sixth Order CMIRK Interpolant

The sixth order discrete formula of the BVP_SOLVER package uses five stages and the local interpolant, $u_i(t)$, a polynomial in θ ($\theta = (t-t_{i-1})/h_i$) of degree at most six, requires three additional stages ($s^* = 8$) with corresponding abscissa, c_6, c_7, c_8 . Its derivative, $u_i'(t)$, is then a polynomial of degree at most five. The free parameters arising in the derivation of $u_i(t)$ were chosen based on an analysis of the continuous local error, as noted above. The associated defect satisfies (as $h_i \rightarrow 0$)

$$\delta(t) = G(t)h_i^6 + O(h_i^7), \quad (9)$$

with

$$G(t) = q_0(\theta)F_0 + q_1(\theta)F_1 + \dots + q_\rho(\theta)F_\rho, \quad (10)$$

where the $q_j(\theta)$'s are polynomials of degree six (that are independent of the problem or h_i), the the F_j 's (elementary differentials) depend only on the problem, and $\rho > 1$ is the number of elementary differentials of seventh order - see [5] for further details. As we will see in the next subsection, when we derive a similar expansion of the defect for the new interpolant, $\tilde{u}_i(t)$, our analysis will show that the corresponding $\tilde{G}(t)$ simplifies considerably and leads to insights into how to select the additional abscissa necessary to define $\tilde{u}_i(t)$ from $u_i(t)$.

From (9) and (10) and the above discussion we see that as $h_i \rightarrow 0$, the value of the defect will approach a linear combination of the $q_j(\theta)$ values, where the coefficients of the linear combination depend on the F_j values. For the i^{th} subinterval, it is therefore impossible to know à priori where the maximum

value of the leading term of the defect will occur. The location of the maximum will vary from subinterval to subinterval depending on the problem. Figure 1 illustrates the typical behavior of the defect in this case.

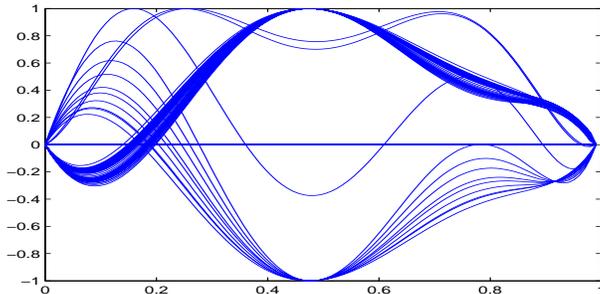


Figure 1: A typical plot of the normalized defect over all subintervals arising in the numerical solution of a BVP by BVP_SOLVER using the standard sixth order CMIRK interpolant. The location of the maximum defect on each subinterval varies from subinterval to subinterval.

3.2 Derivation and Analysis of a New Sixth Order Hermite-Birkhoff Interpolant

In this subsection we demonstrate the application of the boot-strapping algorithm in the development of a new sixth order interpolant, $\tilde{u}_i(t)$, that will be shown to allow an asymptotically correct estimate of the defect for any $t \in [t_{i-1}, t_i]$. We will use this estimate to determine an asymptotically correct estimate of the maximum defect on each subinterval.

As mentioned earlier, the standard computation performed by BVP_SOLVER yields, on the i^{th} subinterval, discrete solution values, y_{i-1} and y_i and an interpolant, $u_i(t)$, defined by the CMIRK formula. Considering the local problem, (2), the y_i and $u_i(t)$ values have local errors that are $O(h_i^7)$. The boot-strapping algorithm will construct a Hermite-Birkhoff interpolant of the form (7). From the form of this interpolant it is easy to see that it depends on approximate solution and derivative values and thus the error associated with this interpolant has contributions from both the data values upon which it is based and from the standard interpolation error. In what follows we will derive an interpolant, $\tilde{u}_i(t)$, that has an error that is dominated by the data error; that is, the degree of $\tilde{u}_i(t)$ will be at most seven and eight data values will be interpolated so that the interpolation error will be one order higher than the data error.

Note that, if we were to have used the bootstrapping process, defining $\tilde{u}_i(t)$ to be a polynomial of degree at most six, using the abscissa c_9, c_{10}, c_{11} (as was done in [18]) then there would be two terms contributing to the leading term in

the expansion of the defect. The first term is $q_0(\theta)F_0h_i^6$, where

$$q_0(\theta) = \frac{d}{d\theta}[\theta^2(1-\theta)^2(\theta-c_6)(\theta-c_7)(\theta-c_8)],$$

$$F_0 = \frac{z_i^{(7)}(t_{i-1})}{7!},$$

and this is the contribution that arises from the inherent interpolation error which is a consequence of our use of a degree six interpolating polynomial to approximate the exact local solution, $z_i(t)$. The second term contributing to the leading coefficient in the expansion of the defect for such a $\tilde{u}_i(t)$ would be directly related to the discrete local error on this subinterval and satisfies

$$\tilde{q}_1(\theta)h_i^6F_1 = \tilde{q}_1(\theta)\frac{[z_i(t_i) - y_i]}{h_i}.$$

Our objective, in deriving a suitable new interpolant, $\tilde{u}_i(t)$, is to have only one term contributing to the leading coefficient in the expansion of the associated defect. To accomplish this we must increase the degree of the resulting $\tilde{u}_i(t)$ (to be at most degree seven) by adding an additional abscissa, c_{12} . This will eliminate the contribution to the inherent interpolation error (i.e., the F_0 term now becomes $O(h_i^7)$).

The derivation of, $\tilde{u}_i(t)$, the special sixth order interpolant that allows an asymptotically correct defect estimate involves several considerations:

1. As mentioned earlier, the standard sixth order interpolant, $u_i(t)$, computed by BVP_SOLVER is of degree six and involves $k_1 = f(t_{i-1}, y_{i-1})$, $k_2 = f(t_i, y_i)$, three stages that were computed for use by the discrete MIRK formula (4) and three additional stages needed for the CMIRK formula (6).
2. The new interpolant will involve the same y_{i-1} , y_i , k_1 and k_2 values, and will use the process of bootstrapping to define four new stages, with corresponding abscissa $c_9, c_{10}, c_{11}, c_{12}$, based on evaluations of the standard interpolant: That is, for $j = 1, 2, 3, 4$,

$$k_{8+j} = f(t_{i-1} + c_{8+j}h_i, u_i(t_{i-1} + c_{8+j}h_i)).$$

Of critical importance in selecting the new abscissa is the size of leading coefficient in the defect expansion for the resulting $\tilde{u}_i(t)$ relative to the size of coefficients of terms in the next highest order of this defect expansion. Assuming that the new interpolant is designed to allow an asymptotically correct defect estimate, we want to choose any remaining free parameters so that the leading term in the expansion of the defect is significantly larger than those arising in the next highest order term. This will improve the range of subinterval sizes over which the estimate remains reliable.

In [21] the choice of suitable abscissa values for any sixth order ‘‘improved’’ interpolant for use with continuous Runge-Kutta formulas is discussed

and the values $\frac{7}{100}, \frac{14}{100}, \frac{86}{100}, \frac{93}{100}$ were found to be close to optimal. We have used these values to define the additional stages of our new local interpolant.

3. Then $\tilde{u}_i(t)$, is the unique polynomial of degree at most seven that satisfies $\tilde{u}_i(t_{i-1}) = y_{i-1}, \tilde{u}_i(t_i) = y_i, \tilde{u}'_i(t_{i-1}) = f(t_{i-1}, y_{i-1}), \tilde{u}'_i(t_i) = f(t_i, y_i)$, and,

$$\tilde{u}'_i(t_{i-1} + c_{8+j}h_i) = k_{8+j} \text{ for } j = 1, 2, 3, 4.$$

This Hermite-Birkhoff representation of this interpolant has the form

$$\begin{aligned} \tilde{u}_i(t_{i-1} + \theta h_i) = & d_0(\theta)y_{i-1} + d_1(\theta)y_i + \\ & h_i \left(\tilde{b}_1(\theta)k_1 + \tilde{b}_2(\theta)k_2 + \tilde{b}_9(\theta)k_9 + \tilde{b}_{10}(\theta)k_{10} + \tilde{b}_{11}(\theta)k_{11} + \tilde{b}_{12}(\theta)k_{12} \right), \end{aligned} \tag{11}$$

4. Since the evaluations of $u(t)$ have an error that is $O(h_i^7)$, the corresponding stages, k_9, \dots, k_{12} , also have an error that is $O(h_i^7)$ (with a Lipschitz assumption on f) and thus the contributions to the error of the Hermite-Birkhoff interpolant from the terms, $h_i k_9, \dots, h_i k_{12}$, are $O(h_i^8)$. A similar argument shows that the contribution from the $h_i k_2$ term is $O(h_i^8)$ (and we note that the y_{i-1} and k_1 terms are “exact” for the local solution and therefore don’t contribute to the data error.)
5. Thus in the new interpolant, $\tilde{u}_i(t)$, there are data error contributions of $O(h_i^8)$ from all terms except the $d_1(\theta)y_i$ term. Furthermore because there are eight data values that are interpolated, standard theory for Hermite-Birkhoff interpolation shows that the interpolation error is $O(h_i^8)$; that is the interpolation error is dominated by the $O(h_i^7)$ data error from the $d_1(\theta)y_i$ term.
6. It then follows that the error expansion for the new interpolant has the form

$$d_1(\theta)F_1 h_i^7 + O(h_i^8)$$

and then the defect satisfies

$$\delta(t) = d'_1(\theta)F_1 h_i^6 + O(h_i^7). \tag{12}$$

(Equation (12) gives us the explicit representation for the $\tilde{q}_1(\theta)$ polynomial mentioned earlier - we have $\tilde{q}_1(\theta) = d'_1(\theta)$.) If $F_1 \neq 0$, the defect for $t \in [t_{i-1}, t_i]$ will always approach a multiple of the polynomial $d'_1(\theta)$ and an asymptotically correct estimate of the maximum magnitude of the defect can be obtained by an evaluation of the defect at the location in the subinterval corresponding to the extreme of $d'_1(\theta)$ for $\theta \in [0, 1]$. That is, as $h_i \rightarrow 0$, *the maximum will occur at the same place within each subinterval, for every problem.* For the above choice of abscissa, plots of the corresponding $q_1(\theta)$ are given in [21] and its local maximum occurs at $\theta = \frac{1}{2}$. Figure 2 illustrates the typical behavior of the defect for this $\tilde{u}(t)$, when h_i is sufficiently small.

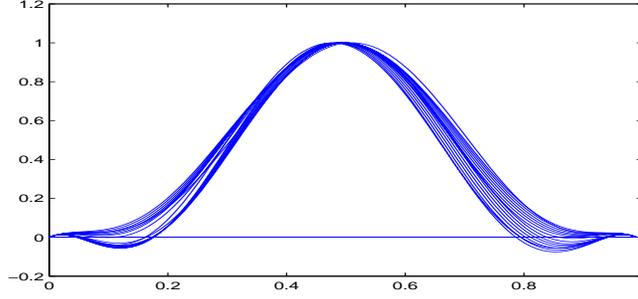


Figure 2: Typical plot of the normalized defect over all subintervals arising in the numerical solution of a BVODE by BVP_SOLVER using the new sixth order interpolant. The location of the maximum defect changes very little from one subinterval to the next, for sufficiently small h_i .

Once the new stages, k_9, \dots, k_{12} , are computed, the new interpolant, $\tilde{u}(t)$, can be evaluated at the point $t_{i-1} + \theta h_i$ by first evaluating the weight polynomials $d_0(\theta)$, $d_1(\theta)$, $\tilde{b}_1(\theta)$, $\tilde{b}_2(\theta)$, $\tilde{b}_9(\theta)$, $\tilde{b}_{10}(\theta)$, $\tilde{b}_{11}(\theta)$, and $\tilde{b}_{12}(\theta)$ at θ , and then computing the expression on the right hand side of (11). For the indicated choices for c_9, \dots, c_{12} , these weight polynomials are

$$\begin{aligned}
 d_0(\theta) &= \frac{1}{2379157} (150000000 \theta^5 - 225000000 \theta^4 + 68955000 \theta^3 + 3022500 \theta^2 + \\
 &\quad 4758314 \theta + 2379157) (-1 + \theta)^2, \\
 d_1(\theta) &= -\frac{1}{2379157} \theta^2 (-4114971 + 67668314 \theta - 359887500 \theta^2 + 668955000 \theta^3 - \\
 &\quad 525000000 \theta^4 + 150000000 \theta^5), \\
 \tilde{b}_1(\theta) &= \frac{1}{1398594579921} \theta (57682725000000 \theta^4 - 116263550000000 \theta^3 + \\
 &\quad 74099888682500 \theta^2 - 16034537281875 \theta + 1398594579921) (-1 + \theta)^2, \\
 \tilde{b}_2(\theta) &= \frac{1}{1398594579921} \theta^2 (-1 + \theta) (57682725000000 \theta^4 - 114467350000000 \theta^3 + \\
 &\quad 71405588682500 \theta^2 - 14105490083125 \theta + 883120980546), \\
 \tilde{b}_9(\theta) &= -\frac{500000}{110488971813759} \theta^2 (25671000000 \theta^3 - 50402285000 \theta^2 + \\
 &\quad 29834968760 \theta - 4700220651) (-1 + \theta)^2, \\
 \tilde{b}_{10} &= \frac{15625}{21384962286534} \theta^2 (145692000000 \theta^3 - 266121140000 \theta^2 + \\
 &\quad 135113668880 \theta - 11988758061) (-1 + \theta)^2,
 \end{aligned}$$

$$\begin{aligned}\tilde{b}_{11}(\theta) &= \frac{15625}{21384962286534} \theta^2 (145692000000 \theta^3 - 170954860000 \theta^2 + \\ &\quad 39947388880 \theta - 2695770819) (-1 + \theta)^2, \\ \tilde{b}_{12}(\theta) &= -\frac{500000}{110488971813759} \theta^2 (25671000000 \theta^3 - 26610715000 \theta^2 + \\ &\quad 6043398760 \theta - 403463109) (-1 + \theta)^2.\end{aligned}$$

4 Numerical Results and Discussion

We have conducted numerous tests to investigate the effectiveness of the new interpolant in providing BVP_SOLVER with a more robust defect estimation procedure. In this section we report on some of these results, providing a numerical comparison of defect estimation based on the standard sixth order CMIRK interpolant vs. the new sixth order Hermite-Birkhoff interpolant. We present results for three test problems chosen from the literature:

- (i) This example is used to illustrate the use of the MUSN solver and is described in [1]. The ODEs are

$$\begin{aligned}y_1'(t) &= \alpha \frac{y_1(t)}{y_2(t)} (y_3(t) - y_1(t)), & y_2'(t) &= -\alpha (y_3(t) - y_1(t)), \\ y_3'(t) &= \frac{1}{y_4(t)} (B - C(y_3(t) - y_5(t)) - \alpha y_3(t) (y_3(t) - y_1(t))) \\ y_4'(t) &= \alpha (y_3(t) - y_1(t)), & y_5'(t) &= -\frac{C}{D} (y_5(t) - y_3(t)),\end{aligned}$$

with boundary conditions,

$$y_1(0) = y_2(0) = y_3(0) = 1, \quad y_4(0) = -10, \quad y_3(1) = y_5(1),$$

and $B = 0.9, C = 1000, D = 10$; α is a parameter to be chosen. The initial guesses provided with the problem description are

$$\begin{aligned}y_1(t) &\equiv y_2(t) \equiv 1, & y_3(t) &= 1 + 8.91t - 4.5t^2, \\ y_4(t) &\equiv -10, & y_5(t) &= 0.91 + 9t - 4.5t^2.\end{aligned}$$

- (ii) This example models swirling flow between two rotating disks and is taken from [1]. It consists of a mixed second/fourth order nonlinear system:

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

where $f'(t), g(t), f(t)$ are proportional to the radial, angular, and axial velocities of the fluid and ϵ is proportional to the viscosity of the fluid. The boundary conditions are

$$f(0) = f(1) = f'(0) = f'(1) = 0, \quad g(0) = \Omega_0, \quad g(1) = \Omega_1,$$

where $\Omega_0 = -1$ and $\Omega_1 = 1$ are angular velocities at the endpoints. The initial guesses are

$$g(t) = 2t - 1, \quad g'(t) = 2, \quad f(t) \equiv f'(t) \equiv f''(t) \equiv f'''(t) \equiv 0.$$

We convert this ODE system to a system of first order equations before treating it with BVP_SOLVER.

- (iii) This example models a shock wave in one-dimensional nozzle flow and is taken from [1]. It consists of a single, second order, nonlinear ODE:

$$y''(t) = \left(\frac{\frac{1}{2} + \frac{\gamma}{2} - \epsilon A'(t)}{\epsilon A(t)} \right) y'(t) - \frac{y'(t)}{\epsilon A(t) y^2(t)} - \frac{A'(t)}{\epsilon A^2(t) y(t)} \left(1 - \frac{\gamma - 1}{2} y^2(t) \right),$$

where t is the normalized downstream distance, $y(t)$ is the normalized velocity, $A(t) = 1 + t^2$, is the area of nozzle at t , $\gamma = 1.4$, and ϵ is proportional to the inverse of Reynolds number. The boundary conditions are $y(0) = 0.9129$ (supersonic flow in the throat) and $y(1) = 0.375$. The initial guesses are

$$y(t) = 0.9129 + (0.375 - 0.9129)t, \quad y'(t) = 0.375 - 0.9129.$$

We convert this ODE to a system of first order equations before treating it with BVP_SOLVER.

For each problem we make a typical choice of parameter value and tolerance setting, as reported below. The determination of the numerical solution of each problem involves computations over several meshes and for each subinterval of each such mesh, we compare the estimated maximum defect based on the standard interpolant with the estimated maximum defect based on the new interpolant. For the standard interpolant, the maximum defect is estimated by sampling the defect at two points within each subinterval. When the new interpolant is employed, the defect is sampled at the point which corresponds to the maximum value of $d'_1(\theta)$ (for the new sixth order interpolant this is at $\theta = \frac{1}{2}$.) We also determine a good estimate of the true maximum defect value over each subinterval by sampling the defect at 101 uniformly distributed points over the subinterval. In the experimental results reported below, *we define the estimate of the maximum defect to be successful if it underestimates the true maximum defect by less than 1%*. We will also report the number of Newton iterations, NI, required by the code to achieve convergence on the given mesh of N subintervals. In each table, %Success \equiv percentage of subintervals of each mesh for which maximum defect estimate is within 1% of the true maximum defect on that subinterval, and NC \equiv no convergence of the Newton iteration. We also report $\sum_j N_j \times NI_j$ where N_j is the number of subintervals in the j th mesh and NI_j is the number of Newton iterations performed to solve the

CMIRK Interpolant			New Interpolant		
N	%Success	NI	N	%Success	NI
9	NC	12	9	NC	9
18	NC	6	18	NC	6
36	0%	6	36	100%	1
37	0%	1	35	100%	1
$\sum_j N_j \times NI_j = 467$			$\sum_j N_j \times NI_j = 469$		

Table 1: Results for Test Problem (i) with $\alpha = 2.2$ and $TOL = 10^{-9}$. The maximum defect estimate obtained using the CMIRK interpolant is never within 1% of the true defect maximum, but using the new interpolant yields a successful estimate on every subinterval. Both versions of BVP_SOLVER employ meshes of approximately the same size and the same number of Newton iterations are required.

CMIRK Interpolant			New Interpolant		
N	%Success	NI	N	%Success	NI
9	NC	6	9	NC	6
18	11%	6	18	89%	6
36	3%	5	36	100%	5
75	0%	1	77	99%	1
89	0%	1	91	99%	1
$\sum_j N_j \times NI_j = 506$			$\sum_j N_j \times NI_j = 510$		

Table 2: Results for Test Problem (ii) with $\epsilon = 10^{-4}$ and $TOL = 10^{-6}$. The maximum defect estimate obtained using the CMIRK interpolant is rarely within 1% of the true defect maximum, but using the new interpolant yields a successful estimate on almost every subinterval. Both versions of BVP_SOLVER employ meshes of approximately the same size and the same number of Newton iterations are required.

nonlinear system defined on that mesh; this summation gives a good machine-independent measure of the relative cost of each computation, as we will explain later in this section.

From Tables 1,2, and 3, we make two general observations:

- We can first observe that the standard interpolant rarely leads to a successful estimate of the defect. Over the problems considered here and over all subintervals of all meshes treated, the estimated maximum defect was within 1% of the true maximum defect for, on average, about 5% of the subintervals. In contrast, the maximum defect estimates from the new interpolant are generally in good agreement with the true maximum defects. This is particularly true on the terminal meshes where the success rate approaches 100%. We observe that the success rates for the new in-

CMIRK			New		
N	%Success	NI	N	%Success	NI
9	NC	6	9	NC	6
18	NC	12	18	NC	12
36	NC	30	36	NC	30
72	11%	30	72	83%	30
181	5%	1	188	97%	1
221	5%	1	228	98%	1
$\sum_j N_j \times NI_j = 3912$			$\sum_j N_j \times NI_j = 3926$		

Table 3: Results for Test Problem (iii) with $\epsilon = 7.5^{-3}$ and $TOL = 10^{-9}$. The maximum defect estimate obtained using the CMIRK interpolant is rarely within 1% of the true defect maximum, but using the new interpolant yields a successful estimate on almost all subintervals. Both versions of BVP_SOLVER employ meshes of approximately the same size and the same number of Newton iterations are required.

terpolant are lower, e.g., 83%, for the first mesh with a converged solution for problem (iii). From a close examination of the computations we observed that the subintervals where estimation failures occurred were ones for which the size of the subinterval was relatively large. In such cases, we are not in the asymptotic regime for the formula and thus the leading term in the expansion does not dominate the higher order terms and one can therefore not expect the one-point sampling process to be valid.

In such cases it is then important to be able to detect the failure of the estimation process and then respond with an alternative strategy for estimation of the maximum defect. In order to allow the code to be able to detect when its estimate of the maximum defect may not be reliable, we are currently investigating an auxiliary process which we call a validity check. This involves a further sampling of the defect on each subinterval at a point where, assuming we are in the asymptotic regime, the value of the defect should be approximately half of what it is at the first sample point. When this validity check fails, we then flag the subinterval as suspect, and can then perform an auxiliary computation to improve the quality of the defect estimate. For example one might sample the defect at several additional points on each subinterval and choose the maximum of these as the estimate of the maximum defect.

- Since the estimates of the maximum defect on each subinterval for the new interpolant are closer to the true maximum defects, they can also be larger, and it may therefore be more demanding (i.e., smaller h_i required) to compute a numerical solution for which the estimated maximum defect is less than the user tolerance. On the other hand, from (12) and (9),(10), we can see that $\tilde{u}(t)$ has only one coefficient, $d'_1(\theta)$, contributing to the

leading term in the expansion of its defect while $u(t)$ has many coefficients contributing to its leading term ($q_j(\theta), j = 0, 1, \dots, \rho$) and therefore the magnitude of the leading term in the expansion of the defect is expected to be smaller for $\tilde{u}(t)$ than it is for $u(t)$. A similar argument applied to the expansion of the respective continuous local errors can be used to show that the size of the local errors associated with $\tilde{u}(t)$ are expected to be smaller than those for $u(t)$ (for the same h_i value.) A second observation we can draw from the above numerical results is that the number of meshes and the number of subintervals per mesh employed by BVP_SOLVER when it uses the new interpolant are about the same as those employed by the code when it uses the standard interpolant. The most significant computational costs incurred by BVP_SOLVER in the numerical solution of a BVODE are associated with the construction and factorization of the large Newton matrices that arise from the discretization of the ODEs on a given mesh, each of which has a cost that is $O(Nn^3)$ for each Newton matrix considered. Thus the sum of terms of the form $N_j \times NI_j$, where j ranges over the meshes employed in the solution of a given problem, gives a good machine-independent measure of the overall computational cost for a given problem. Because the meshes employed by BVP_SOLVER when it uses the new interpolant to get an improved defect estimate are about the same as the ones it uses when it employs the standard interpolant, the overall costs for both computations are comparable. The defect estimation computation is relatively inexpensive - its cost is $O(Nn)$ - and it is performed only after we have obtained a converged discrete solution on a given mesh. An increase in the cost of the defect estimation process by a few function evaluations required to construct the new interpolant does not result in a significant increase in the overall cost.

5 Summary and Conclusions

The BVP_SOLVER code employs discrete and continuous Runge-Kutta formulas of orders two, four, and six, for the numerical solution of BVODEs. For the sixth order case we have derived a new interpolant that allows an asymptotically correct estimate of the defect. The derivation employs a boot-strapping algorithm to obtain the desired interpolant. Numerical results show that the maximum defect estimates for the new interpolant are much more robust than those currently obtained by BVP_SOLVER, based on a CMIRK interpolant. On almost all subintervals of the meshes considered by the code in the numerical solution of the test problems, the computation based on the new interpolant is able to estimate the maximum defect to within 1% of the true maximum defect, while the computation based on the original interpolant is rarely able to obtain this level of success. Furthermore, the numerical results show that the computation based on the new interpolant employs about the same meshes as for the original computation and thus the overall computational costs are comparable to those of the original algorithm. One reason for this is that the better defect

information allows the mesh selection algorithm to do a better job of adapting the mesh to the defect profile. A second reason is that the defect expansion for the new interpolant has a leading expression that involves only one term and thus we can expect that the numerical solutions computed using this interpolant will have a smaller true defect than will the numerical solution computed using the original interpolant.

The more reliable estimates of the maximum magnitudes of the defect we can obtain using our improved interpolants have been used in our modified version of BVP_SOLVER. This change has resulted in a more reliable and robust adaptive mesh strategy that is effective even at coarse meshes where the asymptotic analysis is not applicable. We have also modified the code to implement the Hermite-Birkhoff form of the interpolant in order to obtain a smoother representation of the numerical solution.

Future work involves the development of interpolants that lead to asymptotically correct defect estimates for the more straightforward cases of second and fourth order. More significant future tasks include (i) further investigation of validity checking for the detection of subintervals for which the estimation of the maximum defect may be larger than expected, and (ii) investigation and implementation of new algorithms for effective estimation of the maximum defect on subintervals which are flagged by the validity check and thus for which the asymptotically correct estimate has failed.

References

- [1] 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, Society for Industrial and Applied Mathematics, Philadelphia, 1995.
- [2] U.M. Ascher, J. Christiansen, and R.D. Russell, Collocation software for boundary value ODEs, *ACM Trans. Math. Softw.*, 7 (1981) 209–222.
- [3] G. Bader and U.M. Ascher, A new basis implementation for a mixed order boundary value ode solver, *SIAM J. Sci. Stat. Comp.*, 8 (1987) 483–500.
- [4] K. Burrage, F.H. Chipman, P.H. Muir, Order results for mono-implicit Runge-Kutta methods, *SIAM J. Numer. Anal.*, 31, 1994, 876-891.
- [5] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, Wiley, Chichester, 1987.
- [6] J.R. Cash, A class of implicit Runge-Kutta methods for the numerical integration of stiff differential systems, *J. ACM*, 22, 1975, 540-511.
- [7] J.R. Cash, On the numerical integration of nonlinear two-point boundary value problems using iterated deferred corrections, Part 1: A survey and comparison of some one-step formulae, *Comput. Math. Appl.*, 12a, 1986, 1029-1048.

- [8] J. R. Cash, G. Moore and R. W. Wright, An Automatic Continuation Strategy For The Solution Of Singularly Perturbed Linear Two-Point Boundary Value Problems, *J. Comp. Phys.*, 122, 1995, 266-279.
- [9] J.R. Cash and D.R. Moore, A high order method for the numerical solution of two-point boundary value problems, *BIT*, 20, 1980, 44-52.
- [10] J.R. Cash and A. Singhal, High order methods for the numerical solution of two-point boundary value problems, *BIT*, 22, 1982, 184-199.
- [11] J.R. Cash and A. Singhal, Mono-implicit Runge-Kutta formulae for the numerical integration of stiff differential systems, *IMA J. Numer. Anal.*, 2, 1982, 211-227.
- [12] J.R. Cash and M.H. Wright, A deferred correction method for nonlinear two-point boundary value problems: implementation and numerical evaluation, *SIAM J. Sci. Stat. Comput.*, 12, 1991, 971-989.
- [13] W.H. Enright, A new error control for initial value solvers, *Appl. Math. Comput.*, 31, 1989, 288-301.
- [14] W.H. Enright, Analysis of error control strategies for continuous Runge-Kutta methods, *SIAM J. Numer. Anal.*, 26, 1989, 588-599.
- [15] W.H. Enright, The relative efficiency of alternative defect control schemes for high-order continuous Runge-Kutta formulas, *SIAM J. Numer. Anal.*, 30, 1993, 1419-1445.
- [16] W.H. Enright, Continuous numerical methods for ODEs with defect control, *J. Comp. Appl. Math.*, 125, 2000, 159-170.
- [17] W.H. Enright, K.R. Jackson, S.P. Nørsett, and P.G. Thomsen, Interpolants for Runge-Kutta formulas, *ACM Trans. Math. Softw.*, 12, 1986, 193-218.
- [18] W.H. Enright and W.B. Hayes, Robust and reliable defect control for Runge-Kutta methods, *ACM Trans. Math. Softw.*, 33, 2007, 1-19.
- [19] W.H. Enright and P.H. Muir, Efficient classes of Runge-Kutta methods for two-point boundary value problems, *Computing*, 37, 1986, 315-334.
- [20] W.H. Enright and P.H. Muir, Runge-Kutta software with defect control for boundary value ODEs, *SIAM J. Sci. Comput.*, 17, 1996, 479-497.
- [21] W.H. Enright and L. Yan, The quality/cost trade-off for a class of ODE solvers, submitted to *Numerical Algorithms*, 2008.
- [22] I. Gladwell, L.F. Shampine, L.S. Baca, R.W. Brankin, Practical aspects of interpolation in Runge-Kutta codes, *SIAM J. Sci. Statist. Comput.*, 8, 1987, 322-341.

- [23] S. Gupta, An adaptive boundary value Runge-Kutta solver for first order boundary value problems, *SIAM J. Numer. Anal.*, 22, 1985, 114-126.
- [24] D.J. Higham, Robust defect control with Runge-Kutta schemes, *SIAM J. Numer. Anal.*, 26, 1989, 1175-1183.
- [25] D.J. Higham, Runge-Kutta defect control using Hermite-Birkhoff interpolation, *SIAM J. Sci. Statist. Comput.*, 12, 1991, 991-999.
- [26] J. Kierzenka, L.F. Shampine, A BVP solver based on residual control and the MATLAB PSE, *ACM Trans. Math. Softw.*, 27, 2001, 299-316.
- [27] P.H. Muir, Optimal discrete and continuous mono-implicit Runge-Kutta schemes for boundary value ODEs, *Adv. Comp. Math.*, 10 (1999) 135–167.
- [28] P.H. Muir and B. Owren, Order barriers and characterizations for continuous mono-implicit Runge-Kutta schemes, *Math. Comp.*, 61, 1993, 675-699.
- [29] The Numerical Algorithms Group, Oxford, UK
- [30] R.D. Russell and J. Christiansen, Adaptive mesh selection strategies for solving boundary value problems, *SIAM J. Numer. Anal.*, 14, 1978, 59-80.
- [31] L.F. Shampine, Interpolation for Runge-Kutta methods, *SIAM J. Numer. Anal.*, 22, 1985, 1014-1027.
- [32] L.F. Shampine, Solving ODEs and DDEs with residual control, *Appl. Numer. Math.*, 52, 2005, 113-127.
- [33] L.F. Shampine and P.H. Muir, Estimating conditioning of BVPs for ODEs, *Math. Comput. Modelling*, 40, 2004, 1309-1321.
- [34] L.F. Shampine, P.H. Muir, and H. Xu, A user-friendly Fortran BVP solver, *J. Numer. Anal. Ind. Appl. Math.*, 1, 2006, 201-217.
- [35] R. Weiss, The application of implicit Runge-Kutta and collocation methods to boundary value problems, *Math. Comp.*, 28, 1974, 449-464.
- [36] K. Wright, Some relationships between implicit Runge-Kutta, collocation Lanczos τ methods, and their stability properties, *BIT*, 10, 1970, 217-227.